

Trabajo Fin de Grado

Grado en Ingeniería de las Tecnologías de Telecomunicación

Desarrollo de aplicación Android de rutinas de entrenamiento y compartición de archivos a través de WiFi Direct

Autor: David Garrido Lázaro

Tutora: Dra. María Teresa Ariza Gómez

**Departamento de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2018



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Desarrollo de aplicación Android de rutinas de entrenamiento y compartición de archivos a través de WiFi Direct

Autor:
David Garrido Lázaro

Tutora:
Dra. María Teresa Ariza Gómez
Profesor titular

Departamento de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2018

Trabajo Fin de Grado: Desarrollo de aplicación Android de rutinas de entrenamiento y compartición de archivos a través de WiFi Direct

Autor: David Garrido Lázaro

Tutora: Dra. María Teresa Ariza Gómez

El tribunal nombrado para juzgar el Trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

Agradecimientos

La elaboración y desarrollo de este Trabajo Fin de Grado no habría sido posible sin el apoyo y el respaldo de muchas personas que me han ayudado en este largo recorrido que es el Grado en Ingeniería de las Tecnologías de Telecomunicación. Es por eso que deseo expresarles mi gratitud a dichas personas en este breve prólogo.

En primer lugar, y como no podría ser de otra manera, a mis padres. Gracias por estar siempre a mi lado, apoyándome en cada caída y ayudándome a volverme a levantar. Gracias por creer en mí y darme palabras de ánimo y consolación, cuando las cosas no salían como estaban previstas. Y, sobre todo, gracias por enseñarme el valor del esfuerzo y que todo sacrificio tiene al final su recompensa.

También quiero agradecerles a mi hermana y a mi hermano por estar siempre a mi lado, y por ser ejemplos a seguir con su dedicación y esfuerzo en todos los ámbitos de sus vidas, tanto personales como profesionales.

Asimismo, me gustaría darle las gracias a María Teresa Ariza Gómez, mi tutora de este Trabajo Fin de Grado y profesora de varias asignaturas de la carrera, por enseñarme tanto y darme la oportunidad de profundizar en el mundo de la programación Android, por sus tutorías y correcciones, sin la cual no habría sido posible abarcar este proyecto.

No quiero olvidar tampoco a todos los compañeros de clase y amigos con los que he compartido esta difícil pero satisfactoria etapa de mi vida, los cuales la han hecho mucho más llevadera. En especial a mi amiga Laura García Corredera, por todos esos momentos que hemos vivido dentro y fuera de la facultad, estudiando en la biblioteca los fines de semana, comiendo a base de tupper; por ese descansito que hacía más ameno el estudio, por aguantarme mis agobios en época de exámenes y también por tus desmayos en los laboratorios de la ETSI. Gracias también por esos momentos en la Alameda, por las risas, por nuestras caras que lo dicen todo sin mediar palabra y por las anécdotas. Gracias por estar siempre ahí.

Y, por último, gracias a todas aquellas personas que en algún momento de esta etapa me dieron su apoyo y confianza.

Gracias.

David Garrido Lázaro
Sevilla, 2018

Resumen

Hoy en día, los teléfonos móviles son mucho más que un simple instrumento de comunicación hablada. Con la llegada de los smartphones, se han incrementado las utilidades de los móviles, dando lugar a cientos de aplicaciones de diferente índole: juegos, fotos, videos, redes sociales y de deporte, entre otras.

En este trabajo se ha desarrollado una aplicación móvil de rutinas de entrenamientos, para que así cualquier usuario que desee hacer deporte sin necesidad de un profesional ni un centro deportivo pueda hacerlo. Además, brinda la oportunidad de compartir los propios vídeos de ejercicios con otros usuarios, haciendo así una experiencia única y diferente a las propuestas en el mercado en la actualidad.

Abstract

Nowadays, mobile phones are much more than a simple instrument of spoken communication. With the arrival of smartphones, mobile utilities have increased, offering hundreds of different applications: games, photos, videos, social networks and sports, among others.

In this project, a mobile application of training routines has been developed, so that any user who wishes to play sports without the need of a professional or a sports center can do so. In addition, it provides the opportunity to share the videos of exercises with other users, thus making a unique experience different from the ones currently on the market.

Índice

Agradecimientos	7
Resumen	9
Abstract	11
Índice	13
Índice de Figuras	16
1 Introducción	20
1.1 Motivación	20
1.2 Objetivos	20
1.3 Antecedentes	21
1.3.1 Aplicación Android de rutinas de entrenamiento adaptadas al usuario usando SQLite y JSON	21
1.3.2 Aplicación Android de grupos de entrenamiento P2P	22
1.3.3 100 Pushups	22
1.3.4 Abdominales Reto de 21 Días	23
1.3.5 Ejercicios en casa	24
1.4 Descripción de la solución propuesta	25
1.4.1 Objetivos	25
1.4.2 Funcionalidades	25
1.4.3 Arquitectura	26
1.5 Estructura de la memoria	27
2 Tecnologías empleadas	29
2.1 Android	29
2.1.1 Historia	29
2.1.2 Características	30
2.2 Java	30
2.2.1 Características	31
2.3 YouTube	31
2.4 WiFi Direct	32
2.5 Android Studio	34
2.5.1 Características	35
2.5.2 Estructura del proyecto	35
2.6 Account Kit	36
2.7 SQLite	37
2.7.1 Características	38
3 Interfaz de usuario	40
3.1 Introducción	40
3.2 Splash Screen	41
3.3 Pantalla de inicio	41
3.4 Registro	43
3.5 Menú	45
3.6 Entrenamientos	46

3.6.1	Ejercicio	46
3.7	<i>Compartir</i>	47
3.8	<i>YouTube</i>	51
3.9	<i>Datos de usuario</i>	52
3.10	<i>Cerrar sesión</i>	56
4	Modelado de la información	58
4.1	<i>Base de datos SQLite</i>	58
4.2	<i>Patrón MVC</i>	58
4.2.1	Modelo	59
4.2.2	Vista	60
4.2.3	Controlador	61
5	Análisis	63
5.1	<i>Diagramas de casos de uso</i>	63
5.1.1	Identificación de actores	63
5.1.2	Casos de uso	64
5.2	<i>Diagramas de clases</i>	67
6	Conclusiones	71
6.1	<i>Posibles mejoras</i>	71
6.2	<i>Conclusiones</i>	71
7	Anexo A: Manual de Instalación	74
8	Anexo B: Account Kit	80
	Bibliografía	85

Índice de Figuras

Figura 1.1 Icono de 100 Pushups.	22
Figura 1.2 Interfaz de usuario de 100 Pushups.	23
Figura 1.3 Icono de Abdominales Reto de 21 Días.	23
Figura 1.4 Interfaz de usuario de Abdominales Reto de 21 Días.	24
Figura 1.5 Icono de Ejercicios en casa.	24
Figura 1.6 Interfaz de usuario de Ejercicios en casa.	25
Figura 1.7 Arquitectura.	26
Figura 2.1 Logo de Android	29
Figura 2.2 Logo de Java.	31
Figura 2.3 Logo de YouTube.	31
Figura 2.4 Logo de WiFi Direct.	32
Figura 2.5 Logo de Android Studio.	34
Figura 2.6 Interfaz gráfica de Android Studio.	36
Figura 2.7 Logo de Account Kit.	36
Figura 2.8 Logo de SQLite.	37
Figura 3.1 Icono de Training Together!	40
Figura 3.2 Splash Screen.	41
Figura 3.3 Pantalla inicial.	42
Figura 3.4 Aviso de campo sin completar.	43
Figura 3.5 Registro por SMS.	43
Figura 3.6 Código de ingreso.	44
Figura 3.7 Registro por correo electrónico.	44
Figura 3.8 Menú.	45
Figura 3.9 Entrenamientos.	46
Figura 3.10 Ejercicio.	47
Figura 3.11 Compartir.	48
Figura 3.12 Aviso de WiFi no activado.	48
Figura 3.13 Descubrimiento de dispositivos.	49
Figura 3.14 Conexión de dispositivos.	49
Figura 3.15 Selección de archivo a enviar.	50
Figura 3.16 Archivo enviado.	50

Figura 3.17 Aviso de petición cancelada.	51
Figura 3.18 YouTube.	51
Figura 3.19 Búsqueda en YouTube.	52
Figura 3.20 Datos de usuario.	53
Figura 3.21 Aviso de falta de datos.	53
Figura 3.22 Valor del IMC.	54
Figura 3.23 Tabla del IMC.	55
Figura 3.24 Guardar datos.	55
Figura 3.25 Cerrar sesión.	56
Figura 4.1 Arquitectura.	59
Figura 4.2 Carpeta raw.	60
Figura 4.3 Carpeta layout.	60
Figura 5.1 Identificación de actores.	63
Figura 5.2 Diagramas de casos de uso.	64
Figura 5.3 Caso de uso I - Entrenamientos.	65
Figura 5.4 Caso de uso II - Compartir.	66
Figura 5.5 Caso de uso III – Datos de usuario.	67
Figura 5.6 Diagrama de clases. Parte 1.	68
Figura 5.7 Diagrama de clases. Parte 2.	69
Figura 7.1 Descarga	74
Figura 7.2 Asistente de instalación.	75
Figura 7.3 Componentes a instalar.	75
Figura 7.4 Términos de uso.	76
Figura 7.5 Ruta de instalación.	76
Figura 7.6 Memoria RAM.	77
Figura 7.7 Instalando.	77
Figura 7.8 Descarga SDK.	78
Figura 7.9 Importación del proyecto.	78
Figura 8.1 Elección del sistema operativo.	80
Figura 8.2 Código para conseguir el keyhash.	81
Figura 8.3 Nombre del paquete y keyhash.	81
Figura 8.4 Identificador de la aplicación.	82
Figura 8.5 Añadir el identificador en string.xml.	82
Figura 8.6 Identificador de acceso del cliente.	82

Figura 8.7 Añadir el identificador del cliente en string.xml.	82
Figura 8.8 Importación del SDK.	83
Figura 8.9 Añadir meta-data en AndroidManifest.xml.	83

1 INTRODUCCIÓN

Este primer capítulo presenta los distintos motivos por el cual ha sido elaborado el presente trabajo y el alcance de la solución propuesta al mismo. Se exponen también los diversos objetivos que se han considerado a la hora de realizar el proyecto, así como los antecedentes similares a la aplicación desarrollada.

1.1 Motivación

Hoy en día, llevar un estilo de vida sano y saludable es una de las inquietudes más reiteradas de la sociedad moderna. Según la Organización Mundial de la Salud (OMS): “La salud es un estado de máximo bienestar físico, psíquico y social, y no solamente la ausencia de enfermedad”. Por tanto, todos los elementos que influyen en nuestra salud son analizados de manera minuciosa, no sólo la actividad física, sino la dieta alimentaria, el descanso e incluso hasta la calidad del aire; todo sea por tener nuestro cuerpo en las condiciones óptimas.

En estos últimos años se ha podido apreciar un auge en el sector deportivo, sobre todo en lo referente a gimnasios y centros de formación física, debido a que el ejercicio físico sigue siendo uno de los factores más importantes e inmediatos a la hora de mantener nuestro cuerpo en forma. Sin embargo, no sólo es posible mejorar nuestra condición física en los centros anteriormente mencionados, ya que el boom tecnológico y el empleo de los smartphones han contribuido al desarrollo de todo tipo de aplicaciones para mantenerse en forma.

Con la aplicación desarrollada en este trabajo se busca, precisamente, acercar al usuario a un estilo de vida saludable, suprimiendo los posibles obstáculos de los centros deportivos, tales como el traslado, el tiempo o el dinero, entre otras; brindándoles así la oportunidad de llevar a cabo distintos entrenamientos de manera fácil e intuitiva.

1.2 Objetivos

Tal y como se ha mencionado anteriormente, el objetivo de la aplicación desarrollada en el presente trabajo es el de dotar al usuario de una herramienta potente y eficaz, a la par que sencilla de manejar, con la cual pueda desarrollar unos hábitos de vida saludable, en lo que a entrenamientos físicos se refiere.

Para ello, la aplicación posee diversas rutinas de ejercicios, centradas en diferentes partes del cuerpo, para poder practicarlas de forma individual, donde cada uno de esos ejercicios cuenta con una detallada descripción para su correcta ejecución.

Por otra parte, la generalidad de las aplicaciones existentes en el mercado, *Play Store* para dispositivos Android y *App Store* para dispositivos iOS, son reproducidas desde un servidor externo, lo cual puede derivar en retrasos, cortes, tiempos de suspensión e incluso la no reproducción del archivo. Como diferencia, los videos de entrenamiento de esta aplicación residen en ella, con lo cual permite que funcione siempre, en cualquier lugar y a tiempo real.

Con el mismo propósito, se ha utilizado la base de datos SQLite, que permite almacenar información en dispositivos empotrados de una forma sencilla, rápida, potente y eficaz. Por tanto, ha sido empleada para guardar tanto los vídeos como la información relativa a los datos del usuario.

Además de todo lo mencionado con anterioridad, la aplicación será capaz de compartir archivos con el resto de los usuarios, ya sean vídeos de entrenamiento, imágenes, ficheros, etc. Con el objetivo de poder distribuir ejercicios propios y así colaborar con el resto de deportistas a conseguir sus objetivos.

1.3 Antecedentes

En este apartado se expondrán brevemente algunas de las diversas aplicaciones móviles desarrolladas en la Escuela Técnica Superior de Ingenieros de Sevilla por diversos alumnos, así como otras ya existentes en el mercado para el entrenamiento personal.

1.3.1 Aplicación Android de rutinas de entrenamiento adaptadas al usuario usando SQLite y JSON

Esta primera aplicación que se presenta trata de un proyecto desarrollado en la Escuela Técnica Superior de Ingeniería de Sevilla por la alumna Miriam Franco Maireles en 2015. El objetivo principal de este trabajo es el de una aplicación de rutinas de entrenamiento que se adapta al usuario usando JSON, pudiendo así ampliar sus ejercicios.

Se trata de una aplicación completa y correctamente ejecutada, y agradezco a Miriam por prestar sus vídeos para la realización de esta aplicación, pero tiene el inconveniente que para añadir nuevos entrenamientos es necesario que tengan un formato determinado, es decir, con una sintaxis específica de JSON y unos parámetros concretos del ejercicio, tales como el nombre, título, subtítulo, descripción, etc.

Además, se trata de una aplicación menos interactiva que la expuesta en este proyecto, debido a que no tiene la opción de compartir los entrenamientos en tiempo real entre varios usuarios, ya que hay que tener las rutinas descargadas con anterioridad, en formato Zip y dentro de la carpeta download, en lugar de poder compartirlo entre usuarios de forma rápida y directa.

1.3.2 Aplicación Android de grupos de entrenamiento P2P

Esta segunda aplicación presentada trata también de un proyecto desarrollado en la Escuela Técnica Superior de Ingeniería de Sevilla por el alumno Ricardo Martín Muñoz en 2016. El objetivo principal de este trabajo es el de una aplicación de entrenamiento en grupo a través de la tecnología P2P, en la que los usuarios realizan deporte en grupo reproduciendo un vídeo en común.

Al igual que en la anterior, en esta aplicación no sólo no se puede compartir los ejercicios entre usuarios, sino que tampoco se puede ampliar el catálogo de entrenamientos, lo que hace que sea una aplicación poco flexible.

A continuación se expondrán de forma concisa algunas de las diversas aplicaciones móviles ya existentes en el mercado para el entrenamiento personal, las cuales no son comparables con el trabajo realizado por una sola persona para un Trabajo Fin de Grado, pero que es interesante observar.

1.3.3 100 Pushups



Figura 1.1 Icono de 100 Pushups.

La aplicación *100 Pushups* ofrece un entrenamiento exclusivo al ámbito de las pushups, o flexiones en español. Dicha aplicación tiene una sobria interfaz, mostrada en la figura 1.2, la cual explica de forma breve cómo hacer el ejercicio. Posteriormente te permite seleccionar el programa que más se aproxima al nivel físico de usuario.

Aunque explica la forma de hacer el entrenamiento y da unas breves pautas para ello, la aplicación en sí es un poco sencilla, sólo se centra en un tipo de ejercicio y apenas describe la forma correcta de hacerlo.



Figura 1.2 Interfaz de usuario de 100 Pushups.

1.3.4 Abdominales Reto de 21 Días



Figura 1.3 Icono de Abdominales Reto de 21 Días.

La aplicación *Abdominales Reto de 21 Días* presenta diversos ejercicios para entrenar la zona central del cuerpo: los abdominales. Para ello, tal y como se representa en la figura 1.4, primero se debe elegir el nivel del entrenamiento, principiante o avanzado. Más adelante ofrece la rutina de entrenamiento para ese día, en el cual está el ejercicio así como su número de repeticiones, o el tiempo en segundos, en su caso.

Esta alternativa es más avanzada que la anterior, debido a que posee una interfaz más trabajada y clara y posee diferentes entrenamientos. Sin embargo, sigue enfocándose solamente en una parte del cuerpo, lo cual suprime la opción de trabajar la totalidad de los músculos.

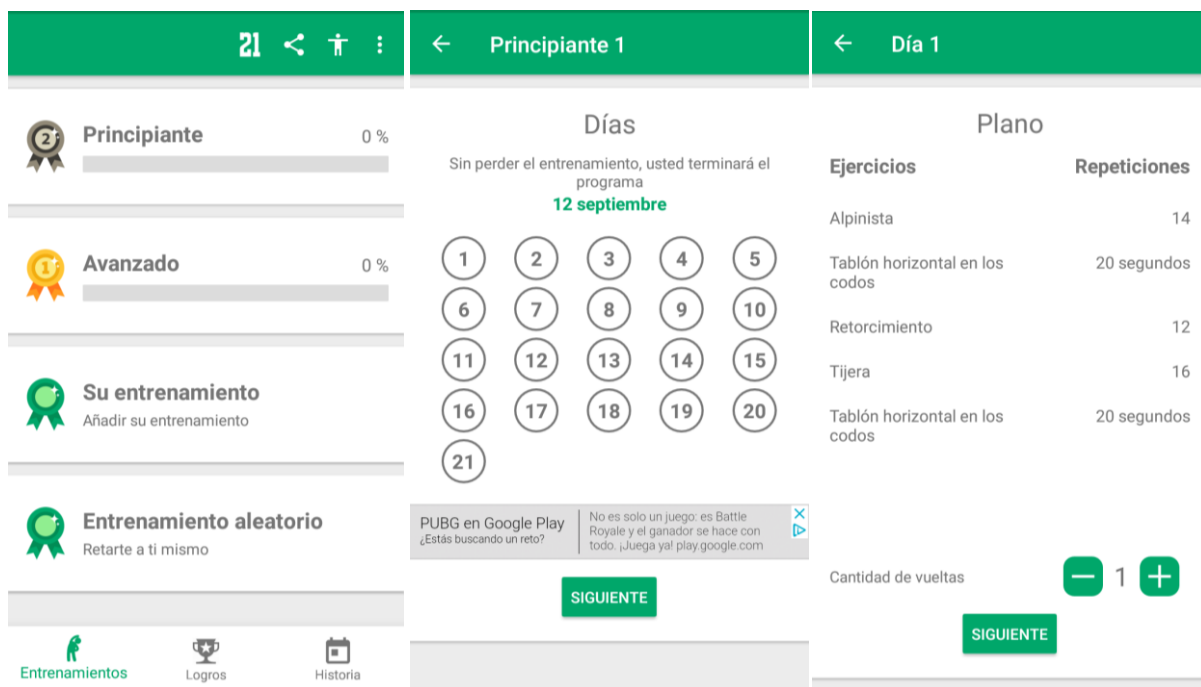


Figura 1.4 Interfaz de usuario de Abdominales Reto de 21 Días.

1.3.5 Ejercicios en casa



Figura 1.5 Icono de Ejercicios en casa.

La aplicación *Ejercicios en casa* ofrece cuantiosos ejercicios para entrenar diversas zonas del cuerpo, lo cual se muestra en la figura 1.6. Para empezar, el usuario decide qué tipo de entrenamiento prefiere realizar y a continuación se le brinda un calendario con los días restantes para superar el desafío, en su caso. El deportista también puede elegir hacer ejercicio libre, si así lo desea.

De las diversas aplicaciones vistas, ésta es la más completa, ya que no sólo presenta ejercicios para distintas partes del cuerpo, sino que también ofrece un breve ejemplo a modo de explicación. Sin embargo, sigue siendo una aplicación con un conjunto cerrado de rutinas, no pudiendo compartir con otros usuarios tus propios entrenamientos.

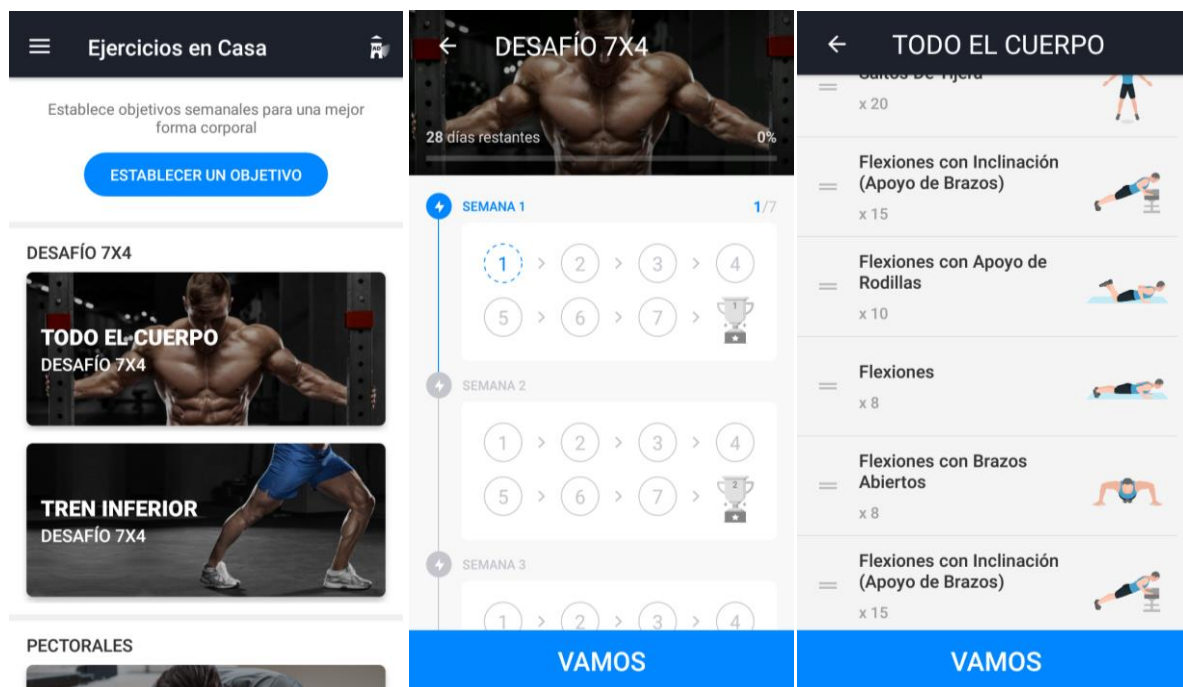


Figura 1.6 Interfaz de usuario de Ejercicios en casa.

1.4 Descripción de la solución propuesta

En este apartado de la memoria del trabajo se expondrá de forma breve las características principales de la solución planteada.

1.4.1 Objetivos

El propósito principal de la aplicación desarrollada es la de dotar a todas aquellas personas que lo deseen con la posibilidad de realizar deporte de forma individual, sin necesidad de un profesional ni un centro deportivo, con la ventaja de poder entrenar en el momento que decidan, sin perder la calidad de los entrenamientos.

Para poder acceder a la aplicación será necesario el registro por parte del usuario, el cual lo podrá realizar por correo electrónico o por SMS a través del móvil. Dicho registro se hará de forma local, quedando así almacenada la información en la base de datos SQLite del propio dispositivo.

1.4.2 Funcionalidades

Una vez se haya iniciado la aplicación, el usuario deberá registrarse, tal y como se ha especificado en el párrafo anterior. Después de esto, el usuario accederá al menú principal, en el cual podrá elegir entre diversas opciones:

- Acceso a las rutinas de entrenamientos incorporadas en la aplicación. Desde ahí se podrá acceder a cada uno de los ejercicios para su realización, cada uno con su descripción para su correcta ejecución.
- Acceso a la sección de *Compartir*, en la cual se pueden enviar y recibir distintos tipos de archivos, para la colaboración y difusión entre deportistas.
- Acceso a YouTube, en el caso de que un usuario desee hacer un entrenamiento que no esté incluido en la aplicación. Así, el usuario podrá buscar y realizar cualquier tipo de ejercicio en dicha plataforma sin necesidad de salir de la aplicación.
- Acceso a los datos del usuario, para que éste pueda modificar su información personal.

Todo esto será explicado con mayor detenimiento en apartados posteriores de la memoria.

1.4.3 Arquitectura

En la figura 1.7 se muestra la arquitectura de la aplicación.

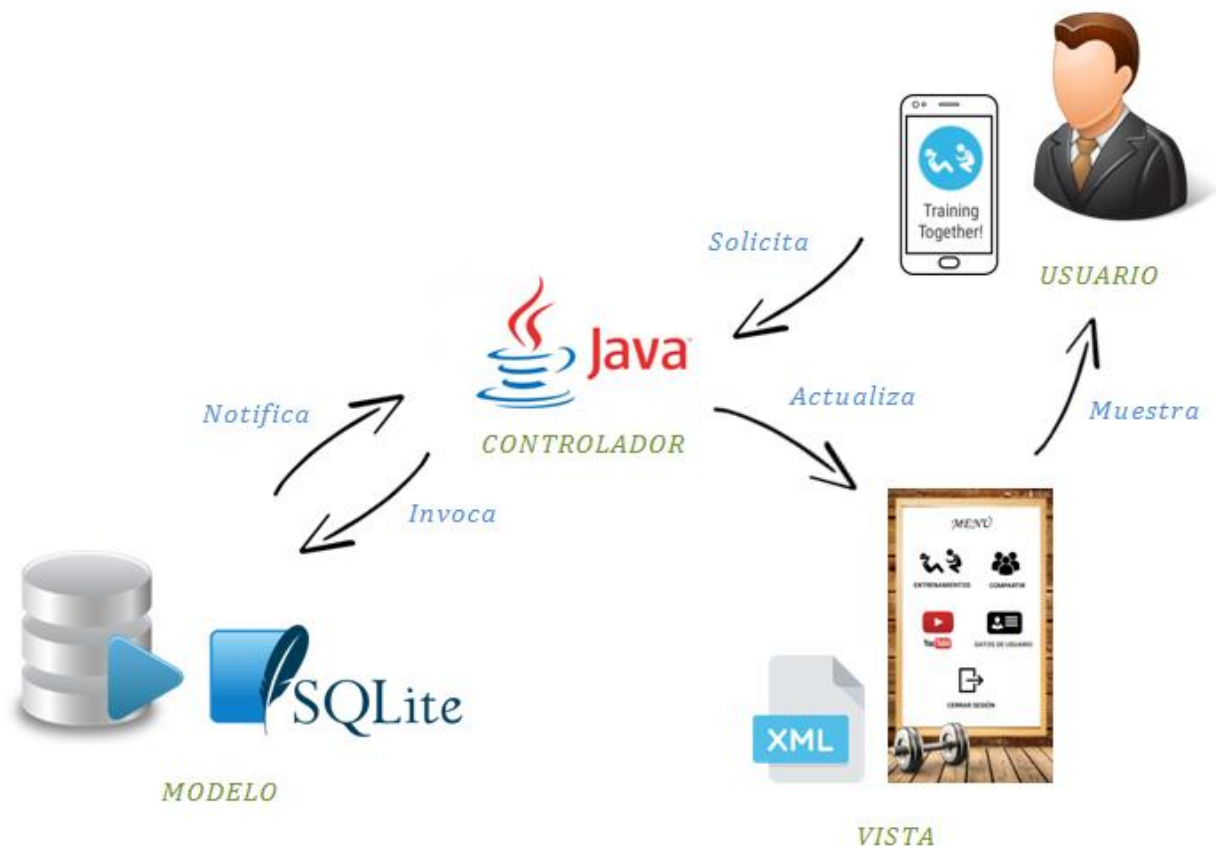


Figura 1.7 Arquitectura.

Como se puede observar en la figura, la aplicación desarrollada sigue el Patrón Modelo-Vista-

Controlador, también llamado Patrón MVC, el cual es un patrón de arquitectura software que utiliza tres componentes para separar los datos y la lógica, es decir, por un lado define los componentes para la representación de la información y por otro lado para la interacción con el usuario:

- Modelo: Es la base de datos SQLite donde se almacena la información del usuario
- Vista: Es la interfaz de usuario con la que se interactúa. Está formada por los layouts, desarrollados en xml.
- Controlador: Es la parte que realiza las operaciones y quien se comunica con el resto de componentes. Además, realiza las consultas a la base de datos SQLite.

1.5 Estructura de la memoria

En este apartado se presenta la estructura que tendrá la memoria de este proyecto, con la finalidad de otorgar al lector una idea general de su organización.

El primer capítulo elabora una pequeña introducción sobre la aplicación y los objetivos que pretende cumplir. Además, enumera alguno de los antecedentes encontrados, así como sus problemas, con el objetivo de exponer una solución propia y factible a dichos problemas.

El segundo capítulo expone las diferentes tecnologías implicadas en la realización de este trabajo, así como sus características principales.

El tercer capítulo muestra con detalle la interfaz de usuario de la aplicación, con cada una de sus funcionalidades y elecciones que puede ejercer el usuario.

El cuarto capítulo realiza una descripción del modelado de la información en la aplicación, en la cual se trata tanto la base de datos SQLite como el Patrón MVC.

El quinto capítulo describe la estructura del software a través de los diagramas UML de casos de uso y de clases.

El sexto y último capítulo comenta algunas de las posibles mejoras del proyecto, las cuales se pueden implementar en futuras líneas de desarrollo, así como las conclusiones a las que ha llegado el autor tras la realización del mismo.

Para concluir, se dispone de un anexo que complementa el contenido del trabajo desarrollado, indicando los pasos a seguir para la instalación de Android Studio, así como la importación del proyecto.

2 TECNOLOGÍAS EMPLEADAS

Este segundo capítulo presenta los distintos entornos de desarrollo empleados para la realización del presente trabajo. Se describirán cada uno de los componentes utilizados, así como los diferentes mecanismos y herramientas implicadas.

2.1 Android

Android, cuyo logo se puede observar en la figura 2.1, es un sistema operativo inicialmente ideado para teléfonos móviles, al igual que iOS, Windows Phone, BlackBerry y Symbian, entre otros. En contraposición a estos, Android está basado en el núcleo de Linux, el cual es libre, gratuito y multiplataforma. Posteriormente ha sido usado también en televisores, relojes inteligentes e incluso automóviles.



Figura 2.1 Logo de Android

2.1.1 Historia

En Octubre de 2003 se fundó Android Inc con el objetivo de desarrollar un sistema operativo para móviles basado en Linux, el cual estuviera al corriente de la ubicación y preferencias del usuario. Sin embargo, en un principio la intención era desarrollar un sistema operativo avanzado para cámaras digitales, pero más tarde se cambió el foco al determinar que el mercado de las cámaras digitales no era lo suficientemente grande. Se redirigirían los esfuerzos a crear un sistema que pudiera competir con Symbian y Windows Mobile.

En 2005, Google compraba Android Inc y empezaba el desarrollo de una plataforma móvil basada en el kernel de Linux. La idea entonces era la de crear un sistema parecido al de BlackBerry, basado en teclado QWERTY. Ese mismo año comienzan a trabajar en la creación de una máquina virtual Java optimizada para dispositivos móviles.

El 5 de noviembre de 2007 se crea la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio, con el objetivo de desarrollar estándares abiertos para móviles. El mismo día se anuncia la primera versión del sistema operativo Android 1.0 Apple Pie. El 23 de septiembre de 2008 se lanzaba el primer teléfono con Android, el HTC Dream / T-Mobile G1.

En 2009 se lanza la versión 1.5 del SDK que incorpora el teclado en pantalla, a finales de ese mismo año se lanza la versión 2.0 y durante el 2010 las versiones 2.1, 2.2 y 2.3. En el año 2010 llegaría la consolidación de Android, como uno de los sistemas operativos móviles más usados.

2.1.2 Características

El sistema operativo Android presenta una serie de características que destacan en comparación a otros sistemas operativos:

- Open source: Es una plataforma de código abierto y de desarrollo libre, lo que permite la modificación de dicho código y la creación de miles de aplicaciones.
- Adaptable al hardware: Esta libertad de código permite adaptar Android a otros dispositivos además de teléfonos móviles, como tablets, GPS, relojes, cámaras, etc.
- Almacenamiento: Android utiliza para sistema de almacenamiento de datos SQLite, que es una base de datos liviana y fácil de manejar.
- Portabilidad: Debido a que las aplicaciones son desarrolladas en Java, es posible que sean ejecutadas en cualquier tipo de CPU.
- Multitarea: Gracias al sistema multitarea inteligente, Android es capaz de gestionar varias aplicaciones abiertas a la vez, dejando en suspensión aquellas que no se utilicen.
- Soportes: Android soporta gran variedad de formatos multimedia, así como para streaming.

2.2 Java

Java es un lenguaje de programación orientado a objetos creado en 1991 y publicado en 1995 por SunMicrosystem, adquirida por Oracle en 2010, con el fin de que los programadores escribieran el código sólo una vez y lo ejecutaran en cualquier dispositivo.

La intención de SunMicrosystem era crear un lenguaje con una estructura y una sintaxis similar a C y C++, aunque con un modelo de objetos más simple y eliminando las herramientas de bajo nivel. El logo de Java se muestra en la figura 2.2



Figura 2.2 Logo de Java.

Las aplicaciones de Java son generalmente compiladas a bytecode, que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

2.2.1 Características

Las principales ventajas de utilizar el lenguaje de programación Java frente a otros tipos de lenguajes son:

- Orientado a objetos: Proporciona un gran control sobre el código y una mejor organización.
- Flexibilidad: Java es un lenguaje especialmente preparado para la reutilización del código, ya sea funciones o adaptarlo a un nuevo entorno.
- Independencia de plataforma: A diferencia de los programas que requieren de versiones específicas para cada sistema operativo, las aplicaciones desarrolladas en Java funcionan en cualquier entorno, dado que no es el sistema quien las ejecuta, sino la máquina virtual.
- Open source: Java es un código abierto, lo que permite la modificación de dicho código, por lo tanto es un lenguaje expandible.

2.3 YouTube

YouTube es una plataforma usada para compartir gran variedad de vídeos, ya sea para verlos como espectador o para subir contenido a modo de productor. La figura 2.3 muestra su logo, el cual fue actualizado en 2017.



Figura 2.3 Logo de YouTube.

YouTube fue creado por tres antiguos empleados de PayPal en Febrero de 2005 y en Octubre de 2006 fue adquirido por Google Inc. YouTube usa un reproductor en línea basado en HTML5, el cual es soportado por la gran mayoría de los navegadores web.

En este proyecto se usa la interfaz de YouTube para poder proveer al usuario de una gran cantidad de vídeos relacionado con el mundo de la actividad física sin necesidad de abandonar la aplicación, lo cual concede mucha flexibilidad.

Conceder esta alternativa al usuario es interesante, debido a que desde aquí puede hallar entrenamientos diferentes a los aportados por la aplicación, lo que le permite beneficiarse de mucha más información y tenerla a su alcance usando una única herramienta.

2.4 WiFi Direct

WiFi Direct es definido por la WiFi Alliance como una «certificación para diferentes dispositivos que soportan cierta tecnología que permite la comunicación directa». Es decir, es un protocolo que permite la conexión entre usuarios para el intercambio de archivos, entre otras cosas. La figura 2.3 muestra su logo.



Figura 2.4 Logo de WiFi Direct.

Las redes WiFi convencionales utilizan dispositivos de control conocidos como puntos de acceso inalámbrico o hotspots. Esos dispositivos realizan tres funciones primarias: son el soporte físico para crear y mantener redes inalámbricas, permiten la comunicación entre los dispositivos de la red, y controlan la conexión y desconexión de dispositivos a la red. Los dispositivos no pueden comunicarse directamente y el punto de acceso siempre hace de intermediario entre ellos.

Por el contrario, WiFi Direct es una norma que permite que varios dispositivos WiFi se conecten entre sí sin necesidad de un punto de acceso intermedio, es decir, los dispositivos WiFi Direct pueden funcionar como cliente o como punto de acceso. Cuando se establece una conexión por primera vez entre dispositivos WiFi Direct, los dispositivos negocian y determinan automáticamente cuál de ellos será el punto de acceso.

Un dispositivo con WiFi Direct activado emite una señal hacia otros artefactos en la sala, haciendo saber que está disponible para una conexión. Los usuarios pueden enviar una invitación o recibirla, para efectuar dicha conexión. Cuando los dispositivos están conectados, se puede empezar a compartir archivos.

Además, si se realiza una comparativa con otras formas de comunicación inalámbrica, como por ejemplo el Bluetooth, se puede observar que WiFi Direct está compuesto por diversas mejoras, entre las que se encuentran:

- Conexiones más rápidas, a una velocidad media de 3.0 Mbps.
- Mayor rango de actuación, hasta los 100 metros.
- Mayor seguridad, debido a la implementación de encriptación WPA2.

En lo referente a la aplicación, el proceso de conexión consta de tres fases:

- Al inicio, la aplicación hace un descubrimiento de los dispositivos cercanos con los que puede establecer una conexión.
- Después, se establece dicha conexión, para lo cual los dispositivos tienen que negociar quién será el punto de acceso y quién el cliente, o en otras palabras, deciden quién es el propietario del grupo.
- Para finalizar, se lleva a cabo el envío del archivo seleccionado para que el receptor pueda disponer de él en su dispositivo.

Para conseguir todo lo anterior existe la clase `WifiDirectBroadcastReceiver`, la cual es la clase principal que notifica diversos eventos WiFi importantes, como:

- `WIFI_P2P_STATE_CHANGED_ACTION`: Indica que se ha percatado de un cambio en el estado del WiFi.
- `WIFI_P2P_PEERS_CHANGED_ACTION`: Indica una variación en la lista de dispositivos disponibles.
- `WIFI_P2P_CONNECTION_CHANGED_ACTION`: Indica que ha cambiado el estado de la conectividad.
- `WIFI_P2P_THIS_DEVICE_CHANGED_ACTION`: Indica que los parámetros del dispositivo han cambiado.

Por otra parte, la clase `WifiDirectActivity` usa las APIs de WiFi Direct para descubrir los dispositivos cercanos y conectarse a ellos. Dichas APIs son asíncronas y usan interfaces para notificar a la aplicación del éxito, `onSuccess()`, o del fracaso, `onFailure()`, de la operación.

Además, también existe la clase `DeviceDetailFragment`, la cual gestiona y administra un peer y permite la interacción con el dispositivo, es decir, configura la conexión de red y transfiere los datos entre dispositivos. Para ello, verifica si se ha accionado el botón de conectar o desconectar, comprobando los permisos y notificando con un aviso de error, en su caso. Accede a la información del dispositivo, como su dirección, estado o si es el propietario del grupo, entre otros. Por último, envía el archivo seleccionado al dispositivo conectado.

2.5 Android Studio

Android Studio, cuyo logo es mostrado en la figura 2.4, es el IDE (Integrated Development Environment, o Entorno de Desarrollo Integrado en español) oficial para el desarrollo de aplicaciones para Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.



Figura 2.5 Logo de Android Studio.

Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, macOS y GNU/Linux. Ha sido diseñado específicamente para el desarrollo de Android.

Estuvo en etapa de vista previa de acceso temprano a partir de la versión 0.1, en mayo de 2013, y luego entró en etapa beta a partir de la versión 0.8, lanzada en junio de 2014. La primera compilación estable, la versión 1.0, fue lanzada en diciembre de 2014. La última versión estable es la 3.0, y fue lanzada en octubre de 2017.

2.5.1 Características

Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan la productividad durante la compilación de aplicaciones para Android, tales como:

- Un sistema de compilación basado en Gradle flexible.
- Un emulador rápido con varias funciones.
- Un entorno unificado en el que se puede realizar desarrollos para todos los dispositivos Android.
- Instant Run para aplicar cambios mientras la aplicación se ejecuta sin la necesidad de compilar un nuevo APK.
- Integración de plantillas de código y GitHub para ayudar a compilar funciones comunes de las aplicaciones e importar ejemplos de código.
- Gran cantidad de herramientas y frameworks de prueba.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
- Compatibilidad con C++ y NDK.
- Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine.

2.5.2 Estructura del proyecto

Los proyectos en Android Studio contienen uno o más módulos con archivos de código fuente y archivos de recursos:

- Módulos de aplicaciones para Android.
- Módulos de bibliotecas.
- Módulos de Google App Engine.

Todos los archivos de compilación son visibles en el nivel superior de Secuencias de comando de Gradle y cada módulo de la aplicación contiene las siguientes carpetas:

- manifests: contiene el archivo AndroidManifest.xml.
- java: contiene los archivos de código fuente de Java, incluido el código de prueba JUnit.
- res: Contiene todos los recursos, como diseños XML, cadenas de IU e imágenes de mapa de bits.

En la figura 2.5 se puede ver un ejemplo de la interfaz gráfica de Android Studio.

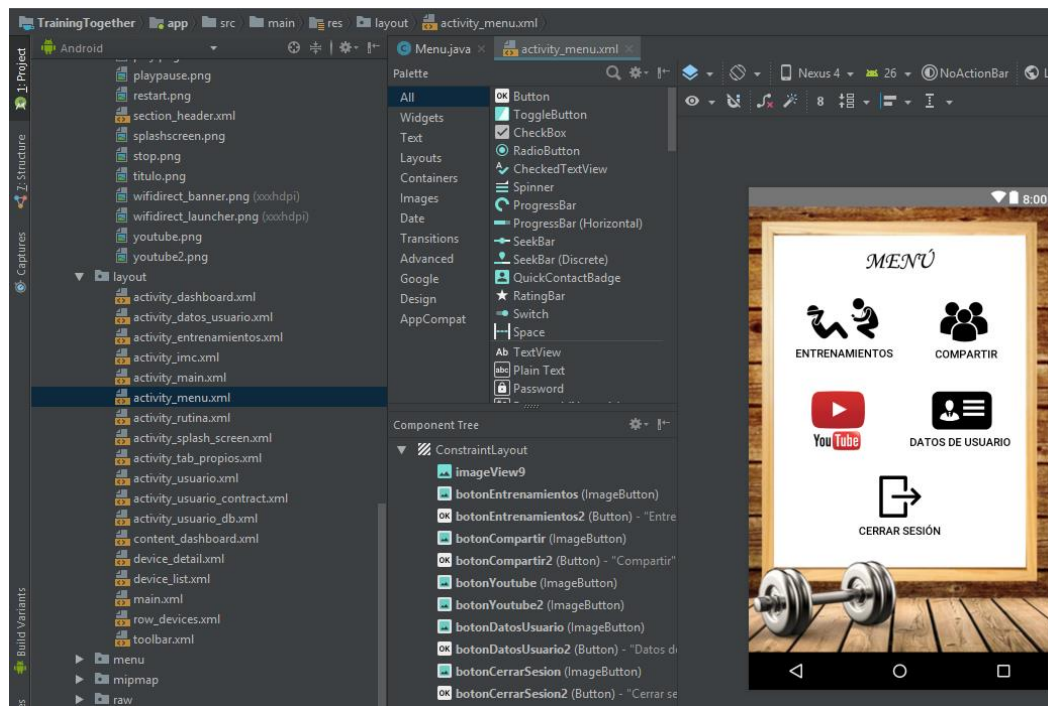


Figura 2.6 Interfaz gráfica de Android Studio.

2.6 Account Kit

Account Kit es una herramienta desarrollada por Facebook que ayuda a las personas a registrarse e iniciar sesión en una aplicación de forma rápida. Es fiable y fácil de usar, y permite elegir cómo registrarse en las aplicaciones. La figura 2.6 representa su logo.



Figura 2.7 Logo de Account Kit.

Account Kit viene a funcionar como una autenticación en dos pasos: el usuario introduce su número de teléfono y recibe un código SMS en su móvil, o bien facilita su correo electrónico y obtiene el código, que debe introducir en la aplicación. Con eso ya se ha iniciado sesión sin necesidad de contraseñas.

Para poder utilizar Account Kit en la aplicación desarrollada, se han tenido que seguir una serie de pautas, las cuales se comentarán brevemente a continuación:

- Importar el SDK de Account Kit.
- Dotar de permisos de Internet en el AndroidManifest.xml.
- Obtener el identificador de Facebook para nuestra aplicación.
- Obtener el identificador de acceso para los clientes.
- Añadir los Hashes de clave de desarrollo y de clave de activación, para poder garantizar la autenticidad de las interacciones entre Facebook y la aplicación.
- Dentro del código:
 - o Verificar si hay un identificador existente válido, para comprobar si ya ha habido previamente una sesión iniciada.
 - o En caso contrario, se comienza un proceso de inicio de sesión, ya sea a través de SMS o correo electrónico.
- Una vez iniciada la sesión del usuario, se redirige hacia el menú principal de la aplicación.
- Proporcionar el cierre de la sesión.

2.7 SQLite

SQLite, donde en la figura 2.7 se puede observar su logo, es una base de datos embebida y muy ligera (~700 kiB). El sistema implementa un motor para la administración de bases de datos relacionales, que permite almacenar información en equipos empotrados de una forma sencilla, eficaz, potente y rápida, en equipos con reducidas capacidades de hardware, tales como los móviles.



Figura 2.8 Logo de SQLite.

A diferencia de los sistemas de gestión de bases de datos convencionales con arquitectura cliente-servidor, el motor de SQLite es independiente, es decir, simplemente se realizan

llamadas a subrutinas o funciones de las propias librerías de SQLite, lo cual reduce ampliamente la latencia en cuanto al acceso a las bases de datos.

2.7.1 Características

Algunas de las características principales de SQLite son:

- **Tamaño:** SQLite tiene una pequeña memoria y una única biblioteca es necesaria para acceder a bases de datos.
- **Costo:** SQLite es de dominio público, y por tanto, se puede utilizar de forma libre para cualquier propósito y también redistribuir libremente.
- **Rendimiento de base de datos:** SQLite realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.
- **Portabilidad:** Se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
- **Estabilidad:** SQLite es compatible con ACID, reunión de los cuatro criterios de Atomicidad, Consistencia, Aislamiento y Durabilidad.
- **Interfaces:** Cuenta con diferentes interfaces del API, las cuales permiten trabajar con C++, PHP, Perl, Python, Ruby, Tcl, groovy, etc.

3 INTERFAZ DE USUARIO

Este tercer capítulo presenta la interfaz de usuario de la aplicación desarrollada. Se explicará detalladamente cada una de las ventanas que lo componen, así como las diferentes opciones que le brinda al usuario.

3.1 Introducción

La aplicación recibe el nombre de *Training Together!*, cuyo logo aparece en la figura 3.1. Una vez instalada en el dispositivo, detallada en el Anexo A, aparecerá en el escritorio del mismo el siguiente icono:



Figura 3.1 Icono de Training Together!

Tal y como ya se mencionó anteriormente, para poder acceder a la aplicación será necesario el registro por parte del usuario, el cual lo podrá realizar por correo electrónico o por SMS a través del móvil.

Una vez se haya iniciado la aplicación, el usuario accederá al menú principal, en el cual podrá elegir entre diversas opciones:

- Acceso a las rutinas de entrenamientos incorporadas en la aplicación. Desde ahí se podrá acceder a cada uno de los ejercicios para su realización, cada uno con su descripción para su correcta ejecución.
- Acceso a la sección de *Compartir*, en la cual se pueden enviar y recibir distintos tipos de archivos, para la colaboración y difusión entre deportistas.
- Acceso a YouTube, en el caso de que un usuario desee hacer un entrenamiento que no esté incluido en la aplicación. Así, el usuario podrá buscar y realizar cualquier tipo de ejercicio en dicha plataforma sin necesidad de salir de la aplicación.
- Acceso a los datos del usuario, para que éste pueda modificar su información personal.

3.2 Splash Screen

Al iniciar la aplicación se muestra durante unos segundos una imagen de bienvenida. A esto se le denomina Splash Screen. La figura 3.2 plasma dicha imagen.



Figura 3.2 Splash Screen.

Este Splash Screen está formado por el logotipo de la aplicación, en el cual aparecen dos personas haciendo deporte, sobre un fondo de color azul.

Además, tal y como se puede observar en el recuadro rojo señalado, la aplicación ocupa la totalidad de la pantalla, permitiendo un aprovechamiento mayor de la pantalla del dispositivo y mejorando así la experiencia del usuario. Esto estará presente en cada una de las interfaces de la aplicación, lo cual se podrá comprobar más adelante.

3.3 Pantalla de inicio

Una vez dentro de la aplicación, se puede observar la interfaz gráfica de la misma. Esta está formada en un primer plano por una pesa sustentada en un suelo de madera, y en un segundo plano por un marco, simulando una pizarra blanca de rotulador en el cual poder escribir los entrenamientos. Todo esto se puede ver en la figura 3.3.

En lo referente a la pantalla de inicio, aparece un saludo de bienvenida, el logo de la aplicación, un recuadro para que el usuario ingrese su nombre y dos opciones para iniciar la sesión: email o SMS.

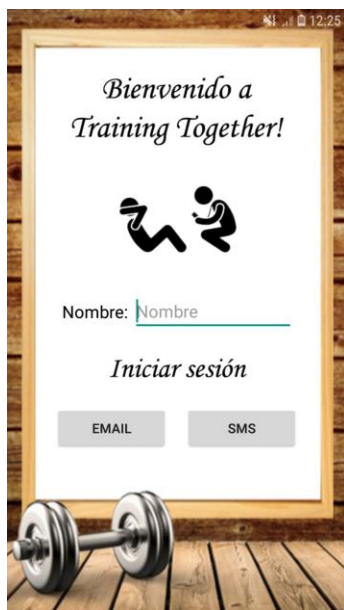


Figura 3.3 Pantalla inicial.

Tal y como se mencionó en el punto anterior, la interfaz de la aplicación ocupa toda la pantalla.

Si el usuario intenta iniciar sesión sin indicar su nombre, con cualquiera de las dos opciones posibles, le aparecerá un aviso, tal y como aparece en la figura 3.4, y hasta que no subsane el error no podrá continuar. En caso contrario, podrá seguir con el registro.

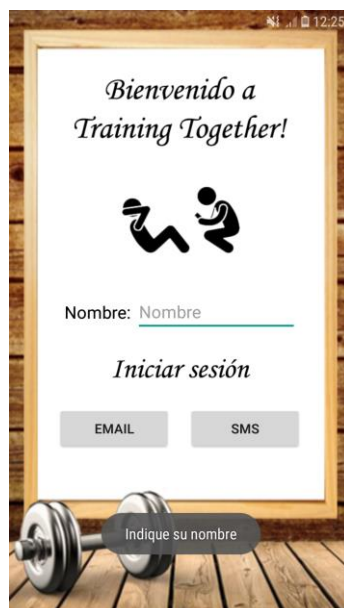


Figura 3.4 Aviso de campo sin completar.

3.4 Registro

Si se elige la opción de registro por SMS, el usuario deberá introducir su número de móvil, como aparece en la figura 3.5.

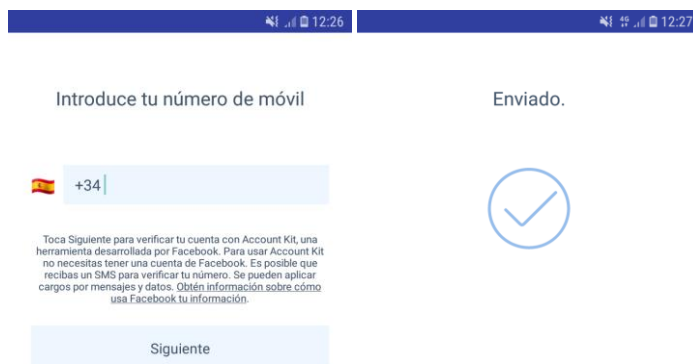


Figura 3.5 Registro por SMS.

A los pocos minutos, se recibirá en el dispositivo un mensaje con un código y el usuario ya podrá acceder a la aplicación, tal como se observa en la figura 3.6 (los datos del usuario han sido ocultados por privacidad).

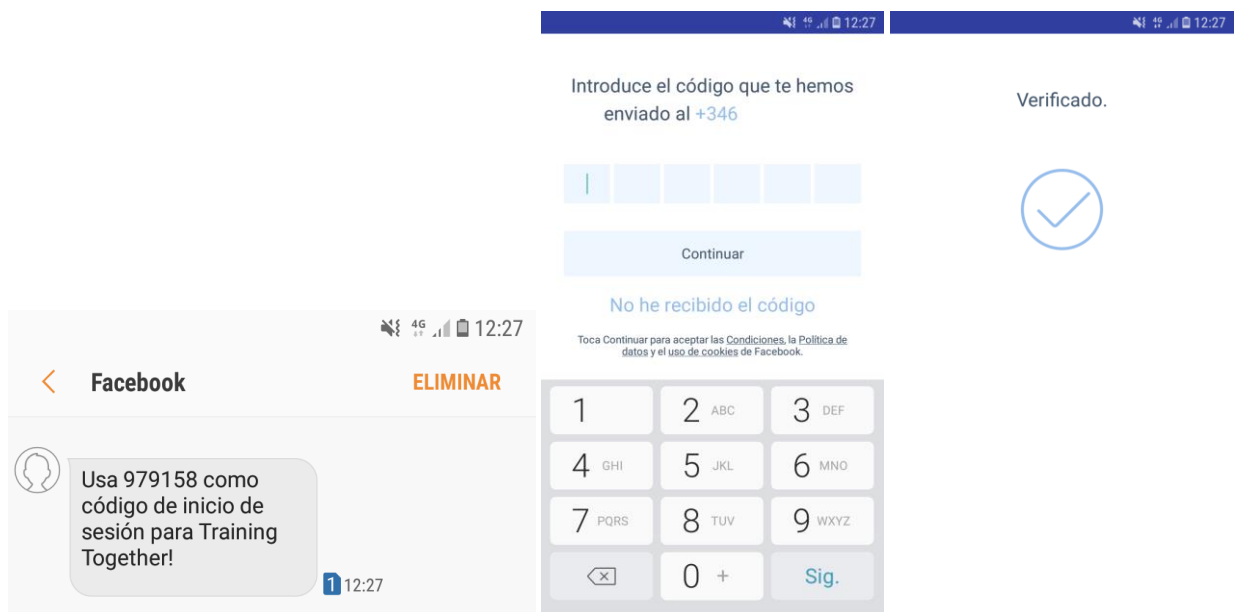


Figura 3.6 Código de ingreso.

Si el usuario prefiere acceder a través del correo electrónico, tendrá que ingresarlo para que pueda ser validado y entrar a la aplicación. La figura 3.7 muestra el proceso (los datos del usuario han sido ocultados por privacidad).

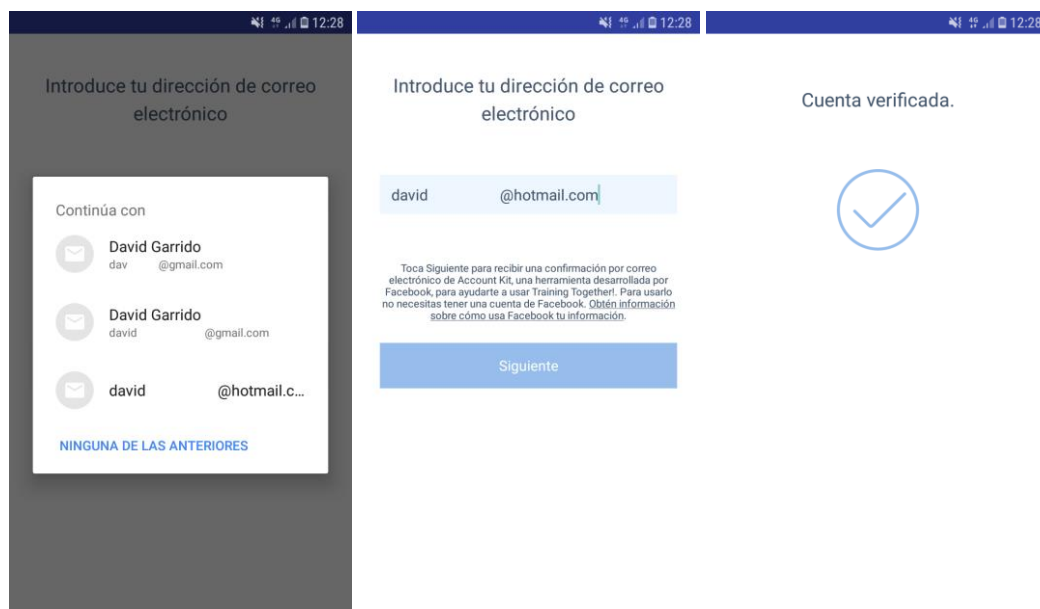


Figura 3.7 Registro por correo electrónico.

3.5 Menú

Una vez que el usuario haya iniciado sesión, aparecerá el menú de la aplicación, mostrado en la figura 3.8, el cual tiene las siguientes opciones:

- Entrenamientos: Acceso a las rutinas de entrenamientos incorporadas en la aplicación. Desde ahí se podrá acceder a cada uno de los ejercicios para su realización, cada uno con su descripción para su correcta ejecución.
- Compartir: Acceso a la sección de compartir, en la cual se pueden enviar y recibir distintos tipos de archivos, para la colaboración y difusión entre deportistas.
- YouTube: Acceso a YouTube, en el caso de que un usuario desee hacer un entrenamiento que no esté incluido en la aplicación. Así, el usuario podrá buscar y realizar cualquier tipo de ejercicio en dicha plataforma sin necesidad de salir de la aplicación.
- Datos de usuario: Acceso a los datos del usuario, para que éste pueda modificar su información personal.
- Cerrar sesión.



Figura 3.8 Menú.

Cada una de las anteriores alternativas será explicada detalladamente en los próximos apartados de la memoria.

3.6 Entrenamientos

En la pantalla de entrenamientos, el usuario podrá ver las diferentes rutinas de entrenamientos integradas en la aplicación. Tal y como se observa en la figura 3.9, cada uno de los ejercicios dispone del nombre de la actividad, una ventana que muestra momentáneamente el ejercicio, una breve descripción y un botón para iniciar el entrenamiento.

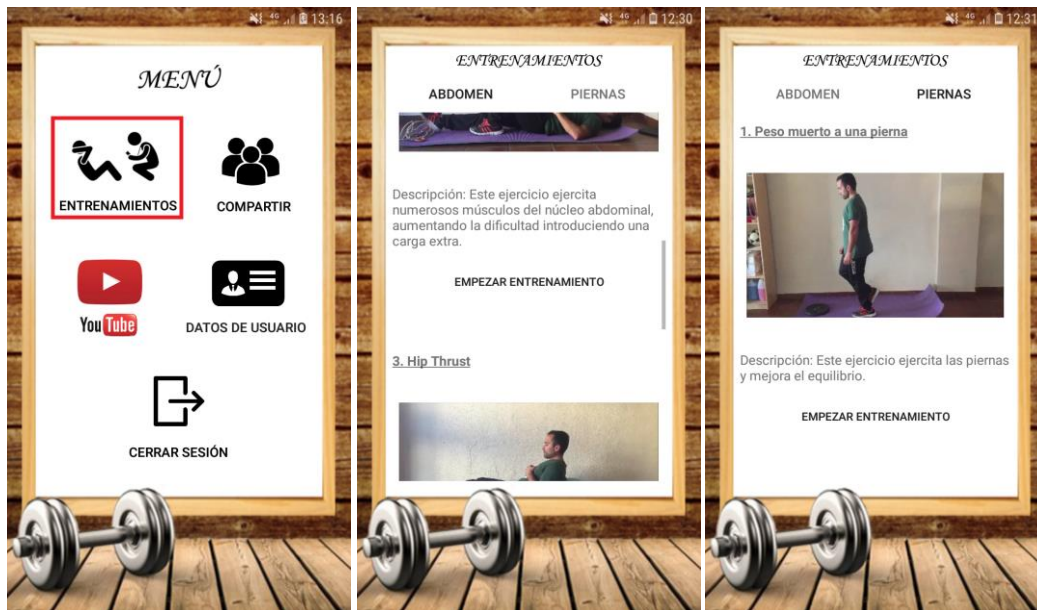


Figura 3.9 Entrenamientos.

Se puede observar en la anterior figura que esta pantalla dispone de un tab o pestaña para cada una de las rutinas, dividiendo el contenido en subcategorías, lo que hace más fácil al usuario encontrar el ejercicio que desea realizar en función de la parte del cuerpo que quiere entrenar.

Además, dispone de un scroll o barra vertical para poder acceder a la totalidad de los entrenamientos incorporados en la aplicación, lo cual la hace mucho más intuitiva.

3.6.1 Ejercicio

Si en cualquier momento el usuario decide hacer algún ejercicio en concreto, solo debe pulsar en “Empezar entrenamiento” y se le abrirá la pantalla mostrada en la figura 3.10.

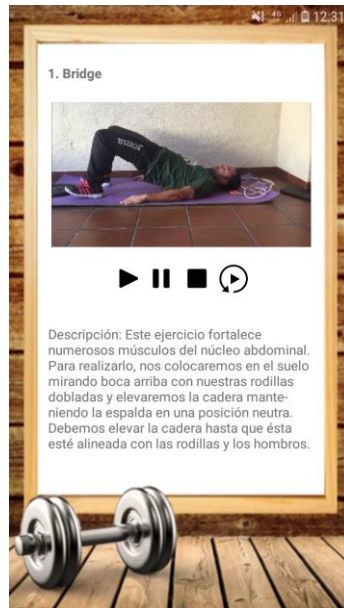


Figura 3.10 Ejercicio.

En esta pantalla, el usuario dispone del nombre del ejercicio, un vídeo demostrativo y una descripción detallada para poder realizarlo correctamente, explicándole paso a paso cómo debe proceder.

Asimismo, también posee cuatro botones para tener el control total sobre el ejercicio:

- Botón de Play: Inicia la primera vez el ejercicio y todas las demás veces en la cual ha sido pausado.
- Botón de Pausa: Detiene momentáneamente el vídeo y después sigue por donde se ha quedado, haciendo uso del botón de Play.
- Botón de Stop: Para por completo el vídeo cuando el usuario ha terminado de hacer el ejercicio. Si se quiere volver a realizar dicho ejercicio hay que hacer uso del botón de Reinicio.
- Botón de Reinicio: Recarga el vídeo desde el principio.

3.7 Compartir

Volviendo al menú, encontramos la opción Compartir, la cual se muestra en la figura 3.11. En esta pantalla aparece en primera instancia el nombre del dispositivo del usuario y el estado de disponible, en su caso. Debajo se muestran los dispositivos disponibles para realizar la conexión, inicialmente ninguno. Por último hay un botón que enuncia “*Descubrir*”.

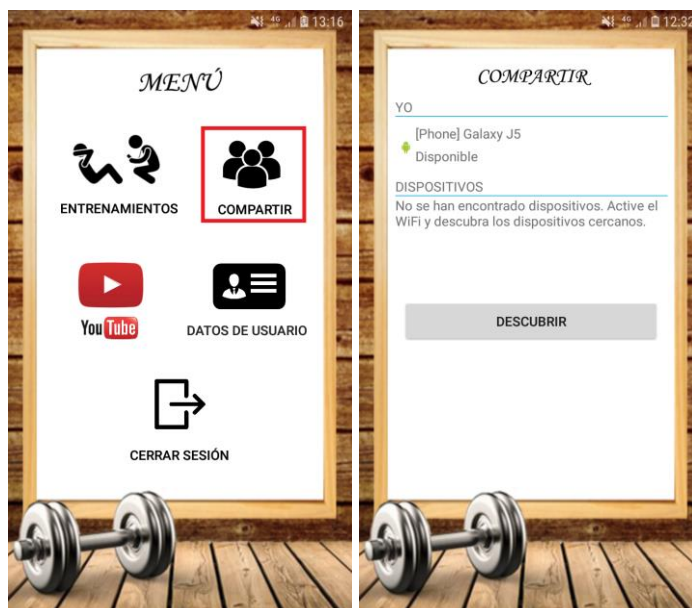


Figura 3.11 Compartir.

Si el usuario desea conectarse con otro individuo, debe presionar el mencionado botón para saber quién está disponible. Sin embargo, si realiza lo anterior sin activar el WiFi le aparecerá el mensaje mostrado en la figura 3.12.

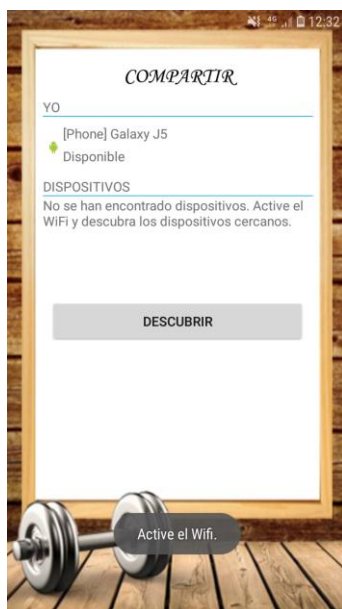


Figura 3.12 Aviso de WiFi no activado.

Una vez activado el WiFi, se podrá proceder a buscar otros dispositivos cercanos. Aparecerá un cuadro de diálogo como el mostrado en la figura 3.13, informando de dicha búsqueda y aparecerá una lista con todos los que haya encontrado. En cualquier momento se podrá anular la búsqueda presionando el botón de atrás.

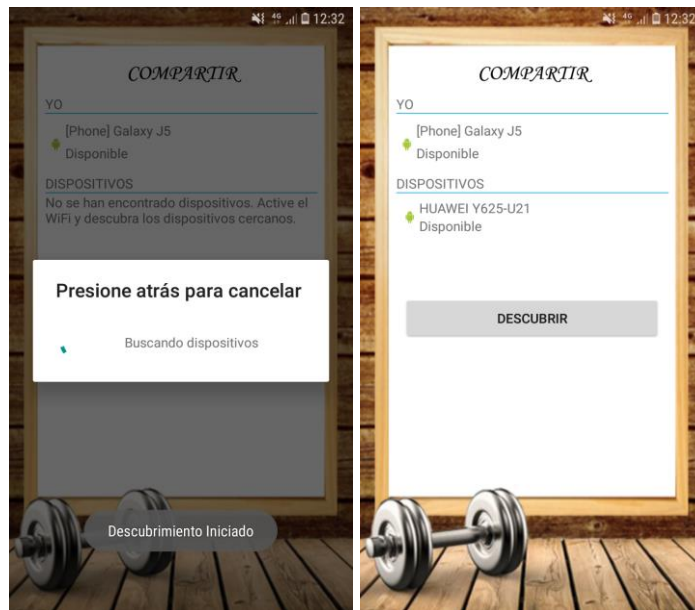


Figura 3.13 Descubrimiento de dispositivos.

Cuando el usuario seleccione el dispositivo al que quiere conectarse, aparecerán dos nuevos botones: Conectar y Desconectar, si previamente ha sido conectado. Al presionar sobre el botón de Conectar, se iniciará la unión entre los dispositivos. En cualquier momento se podrá anular la conexión presionando el botón de atrás. Esto se muestra en la figura 3.14.

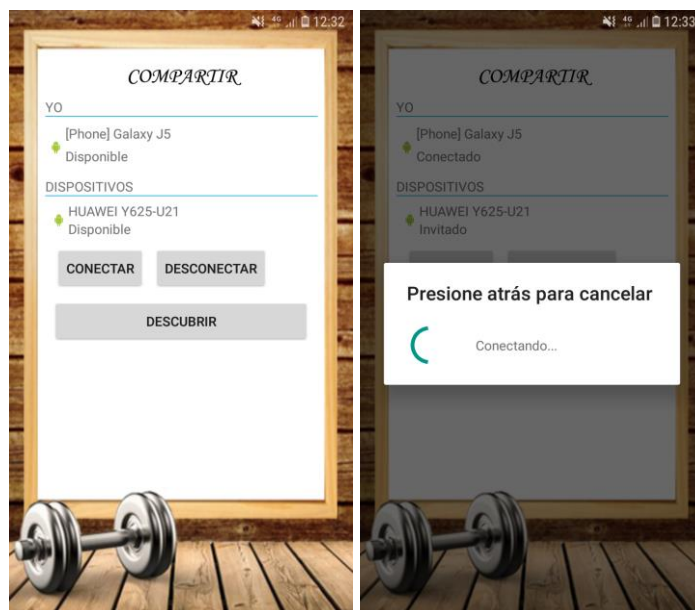


Figura 3.14 Conexión de dispositivos.

Una vez que los dispositivos estén conectados, se puede proceder al envío de ficheros, como aparece en la figura 3.15. Para ello, se pulsa sobre el icono de + que se ha manifestado. Se abrirá el almacenamiento del móvil y el usuario seleccionará el archivo a enviar.

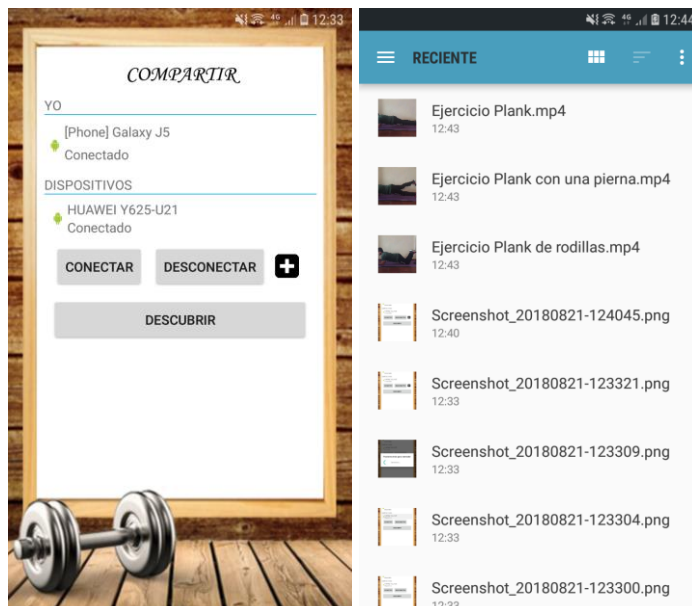


Figura 3.15 Selección de archivo a enviar.

Es entonces cuando la aplicación realizará el envío del archivo hacia el otro dispositivo enlazado. Se mostrará el texto *“Archivo enviado correctamente”*, como se muestra en la figura 3.16, para que el usuario tenga constancia de que el proceso ha sido exitoso.

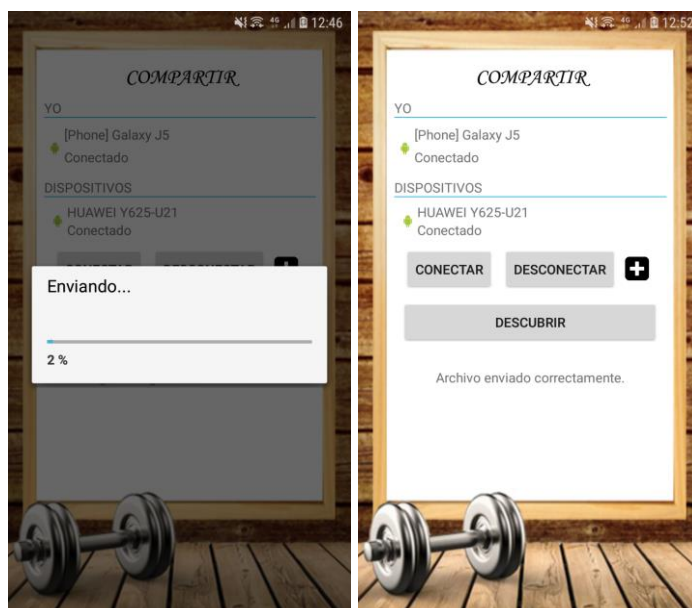


Figura 3.16 Archivo enviado.

Por el contrario, si antes de elegir lo que desea enviar, el usuario presiona el botón de atrás, aparecerá el mensaje de petición cancelada mostrado en la figura 3.17.

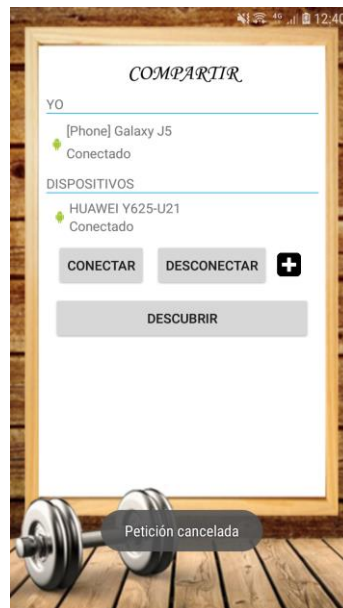


Figura 3.17 Aviso de petición cancelada.

3.8 YouTube

Volviendo al menú, encontramos la opción de YouTube. En esta pantalla, mostrada en la figura 3.18, la aplicación redirige a YouTube, donde el usuario puede expandir su visión del deporte.

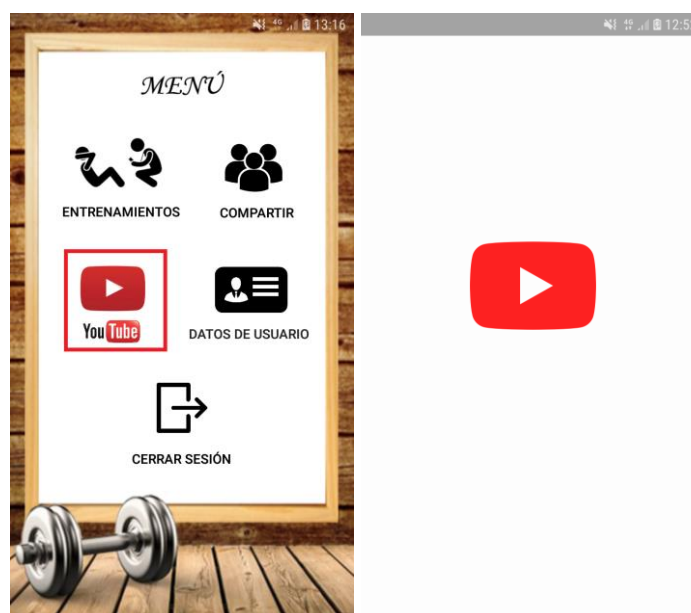


Figura 3.18 YouTube.

Conceder esta alternativa al usuario es interesante a la par que significativo, debido a que desde aquí puede hallar entrenamientos diferentes a los aportados por la aplicación sin necesidad de abandonar la misma, lo que le permite beneficiarse de mucha más información y tenerla a su alcance usando una única herramienta. La figura 3.19 muestra un ejemplo de los ejercicios a los que el usuario puede acceder.



Figura 3.19 Búsqueda en YouTube.

3.9 Datos de usuario

Volviendo al menú, encontramos la opción Datos de usuario, como se muestra en la figura 3.20. En esta pantalla el usuario puede modificar su información personal. En un principio sólo se muestra el nombre del usuario, el cual ingresó al iniciar sesión.

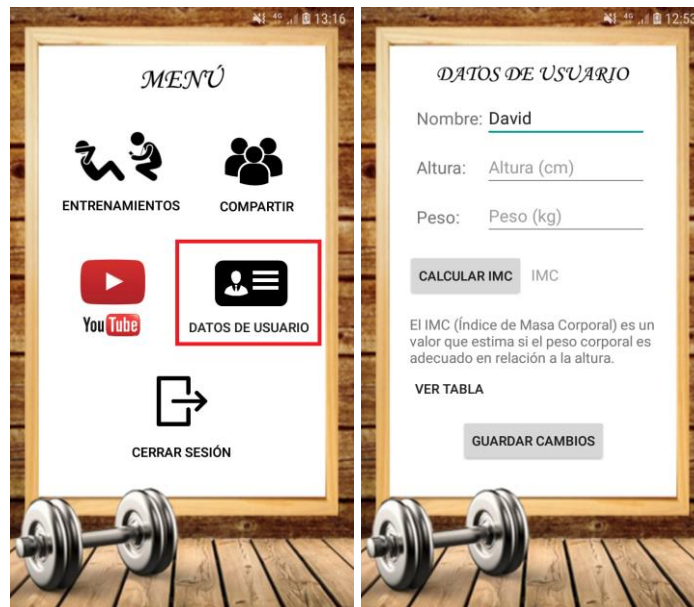


Figura 3.20 Datos de usuario.

Además, el usuario puede calcular su IMC, o Índice de Masa Corporal, que es un valor que estima si el peso corporal de una persona es adecuado en relación a su altura. Para ello, deberá registrar la altura, en centímetros, y el peso, en kilogramos. Si no lo hace, se mostrará un aviso para que indique su altura o peso, en su caso. Esto se muestra en la figura 3.21.

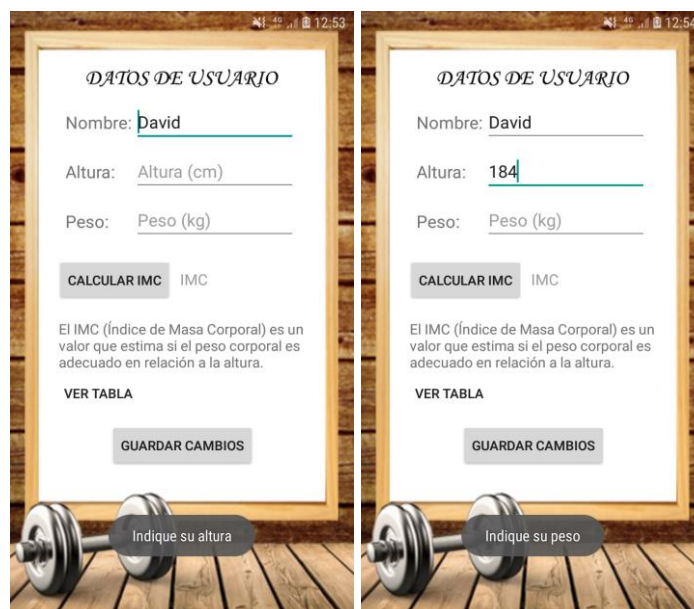


Figura 3.21 Aviso de falta de datos.

Si por el contrario el usuario ha introducido correctamente todos los datos, le aparecerá el valor de su IMC junto con una breve reseña, la cual puede tomar los siguientes valores en función del IMC:

- $IMC < 18.5$: Bajo peso.
- $18.5 < IMC < 24.9$: Peso normal.
- $25 < IMC < 29.9$: Sobrepeso.
- $IMC > 30$: Obesidad.

La figura 3.22 muestra un ejemplo de IMC.

El IMC (Índice de Masa Corporal) es un valor que estima si el peso corporal es adecuado en relación a la altura.

Figura 3.22 Valor del IMC.

Asimismo, si el usuario quiere ver con más detalle el rango de valores para calcular el IMC o desea saber cuánto le falta para llegar a su Índice de Masa Corporal deseado, puede ver una lista más detallada pulsando “*Ver tabla*”, donde aparece una tabla que relaciona el peso y la altura con el IMC, tal como aparece en la figura 3.23.

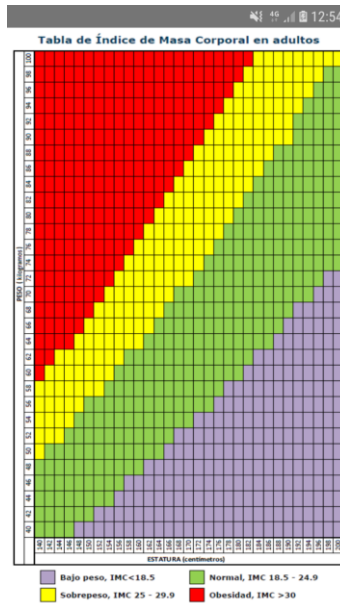


Figura 3.23 Tabla del IMC.

Por último, el usuario puede guardar todos sus datos presionando “*Guardar cambios*”. Gracias a esta funcionalidad, la información quedará guardada para el futuro y podrá observar cómo alcanza sus objetivos y mejora su condición física debido a la aplicación, actualizando los datos tantas veces como sea necesario. Esto se muestra en la figura 3.24.

DATOS DE USUARIO

Nombre:

Altura:

Peso:

CALCULAR IMC 20,09: Peso normal

El IMC (Índice de Masa Corporal) es un valor que estima si el peso corporal es adecuado en relación a la altura.

VER TABLA

GUARDAR CAMBIOS

Datos guardados correctamente

Figura 3.24 Guardar datos.

3.10 Cerrar sesión

Volviendo al menú, encontramos como última opción Cerrar sesión, mostrada en la figura 3.25. Al pulsar en ella, el usuario abandona su sesión abierta previamente y se sale de la aplicación. La sucesiva entrada en la aplicación le llevará de nuevo a la pantalla de inicio de sesión.

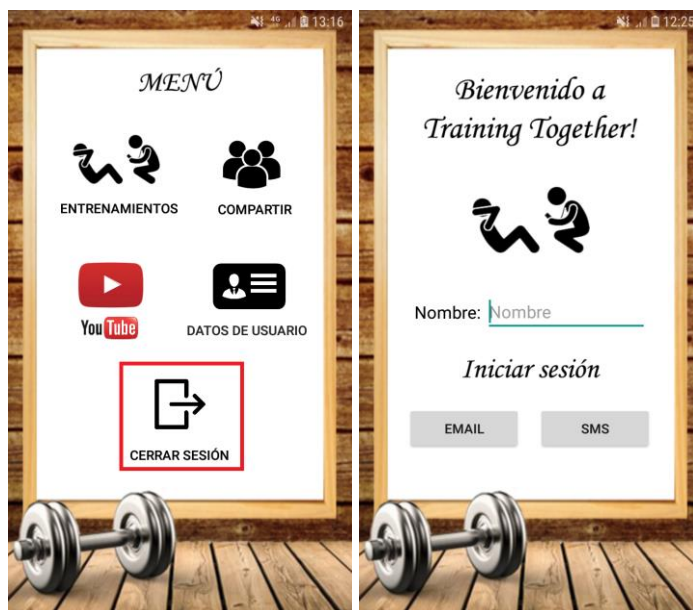


Figura 3.25 Cerrar sesión.

En caso de que el usuario no haga esto, su sesión permanecerá abierta tantas veces entre y salga de la aplicación, sin necesidad de estar ingresando de nuevo sus datos.

4 MODELADO DE LA INFORMACIÓN

Este cuarto capítulo expone el modelado de la información y la arquitectura que conforma la aplicación. Además, se analizará la estructura de la base de datos para ver cómo maneja la información proporcionada por el usuario.

4.1 Base de datos SQLite

La aplicación desarrollada hace uso de SQLite para gestionar la base de datos. Esta base de datos local está compuesta por una única tabla en la que se almacena la información relativa al usuario y consta de los siguientes parámetros:

- ID: Parámetro de tipo INT (entero) utilizada como clave primaria de la tabla. No contiene ninguna información, sino que es usada como organizador de la tabla.
- Nombre: Parámetro de tipo TEXT (texto) que contiene el nombre del usuario ingresado en el registro de la aplicación, o bien modificado posteriormente desde la pantalla de Datos de usuario.
- Altura: Parámetro de tipo INT (entero) que incluye la altura del usuario.
- Peso: Parámetro de tipo INT (entero) que contiene el peso del usuario.

4.2 Patrón MVC

Como ya se mencionó en apartados anteriores, la aplicación desarrollada sigue el Patrón Modelo-Vista-Controlador, también llamado Patrón MVC, el cual es un patrón de arquitectura software que utiliza tres componentes para separar los datos y la lógica, es decir, por un lado define los componentes para la representación de la información y por otro lado para la interacción con el usuario.

Recordemos su arquitectura, mostrada en la figura 4.1.

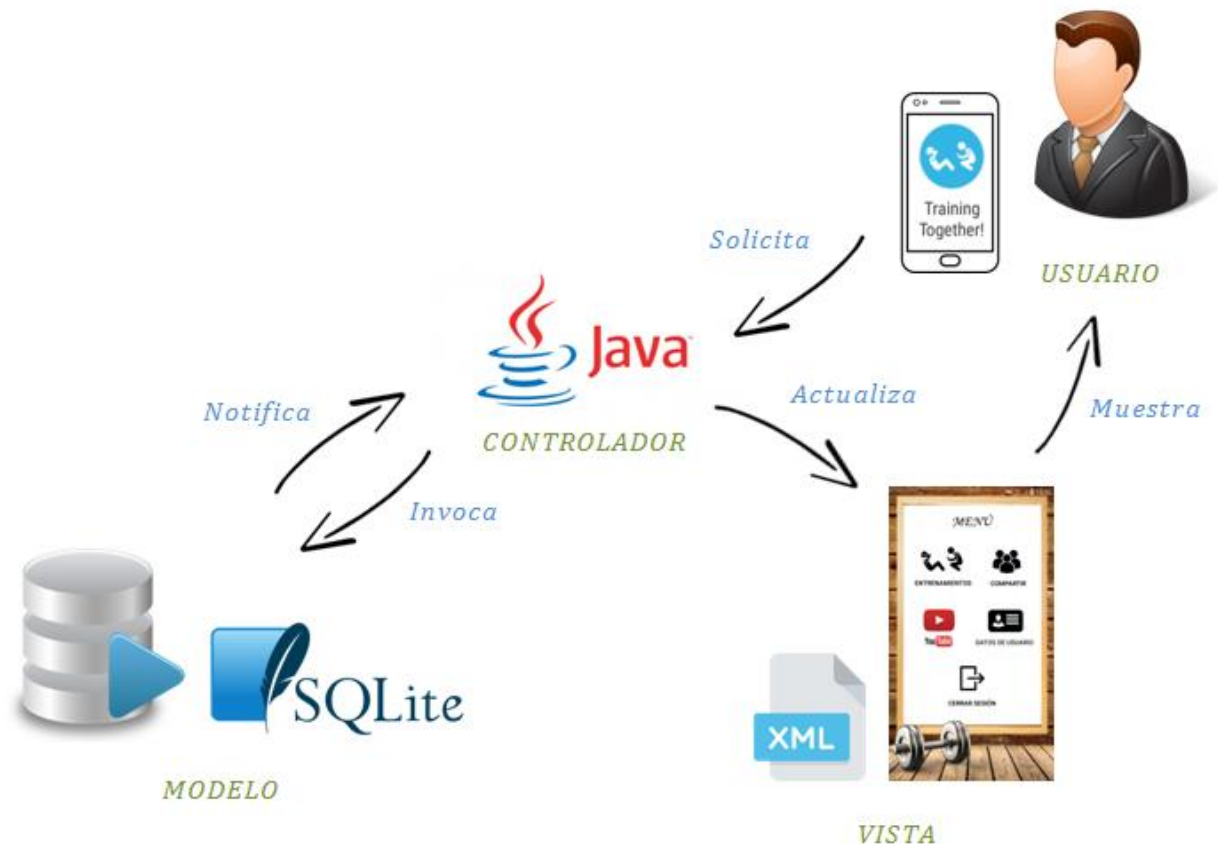


Figura 4.1 Arquitectura.

4.2.1 Modelo

El modelo de la aplicación está formado por dos partes:

- La información dinámica: Son los datos almacenados y variables, como la información del usuario guardada en la base de datos SQLite.
- La información estática: Son los datos residentes en la aplicación desde su instalación, es decir, los vídeos que forman las rutinas de entrenamiento. Están alojados en la carpeta raw, que se muestra en la figura 4.2.

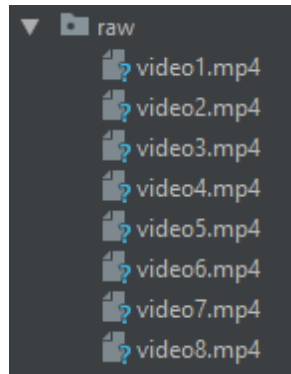


Figura 4.2 Carpeta raw.

4.2.2 Vista

La Vista de la aplicación está formada por las interfaces de usuario con la que se interactúa, es decir, por los layouts desarrollados en xml. Estos son todas aquellas pantallas detalladas en el apartado 3 del presente documento. Están alojadas en la carpeta layout, mostrada en la figura 4.3.

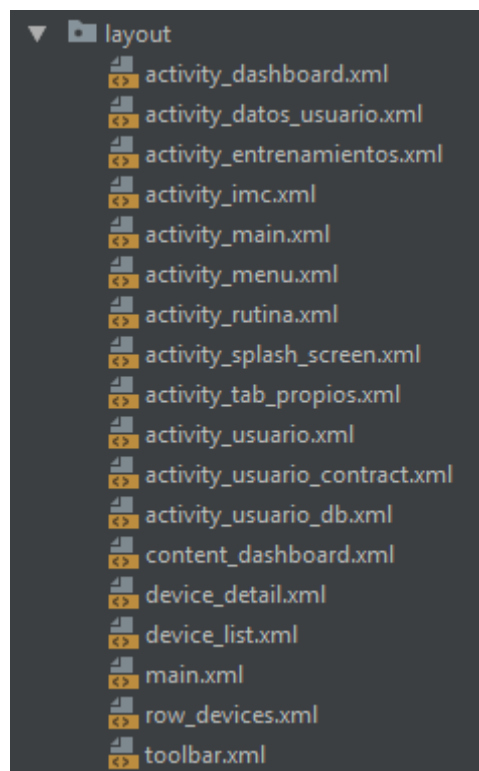


Figura 4.3 Carpeta layout.

Además de lo mencionado anteriormente, la Vista también engloba otro tipo de ficheros gráficos, como son los iconos y el background de la aplicación. Están alojados en la carpeta drawable.

4.2.3 Controlador

Finalmente, el Controlador de la aplicación está formado por las clases que realizan las operaciones y que se comunican con el resto de componentes. Además, realiza las consultas a la base de datos SQLite.

5 ANÁLISIS

Este quinto capítulo expone diversos diagramas para analizar el funcionamiento del sistema de forma detallada en distintos ámbitos. Dichos diagramas se exponen en UML (Unified Modeling Language o Lenguaje Unificado de Modelado, en español). Los diagramas expuestos son los de casos de uso y de clases.

5.1 Diagramas de casos de uso

A continuación se expone una descripción de los casos de uso y sus diagramas. También se especifican los actores que harán uso del sistema con el fin de identificar todos los casos de uso a desarrollar.

5.1.1 Identificación de actores

Usuario: Es el actor que hace uso de la aplicación en su dispositivo, el cual podrá acceder a las funcionalidades del sistema. La figura 5.1 muestra la identificación de actores.

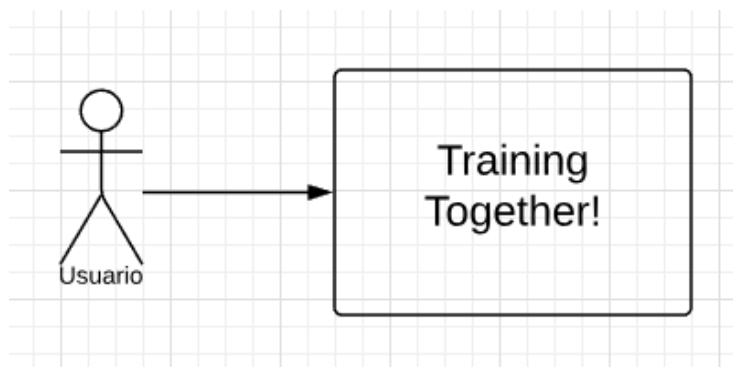


Figura 5.1 Identificación de actores.

5.1.2 Casos de uso

En el diagrama de la figura 5.2 se muestran los casos de uso desarrollados en la aplicación.

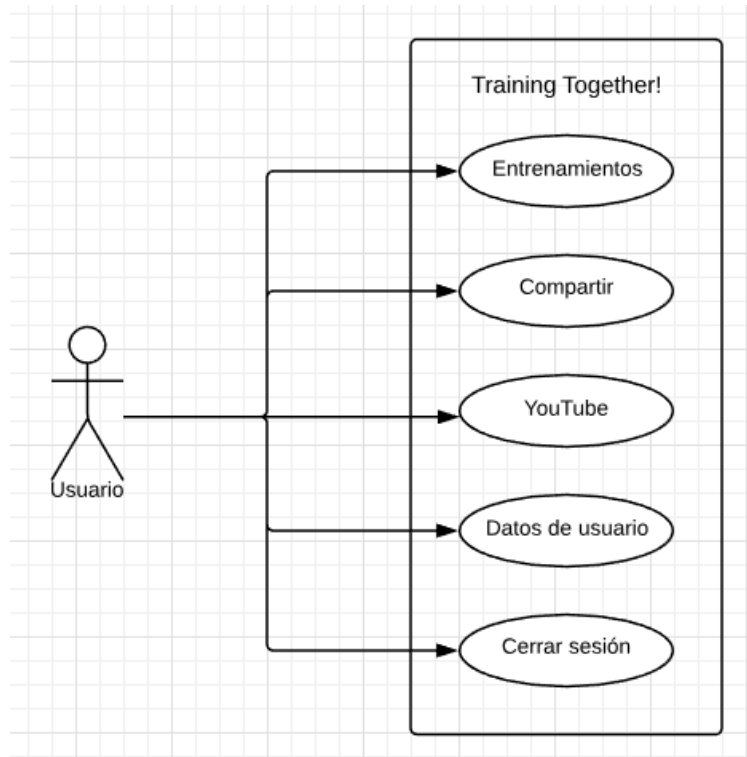


Figura 5.2 Diagramas de casos de uso.

5.1.2.1 Entrenamientos

Este subapartado, mostrado en la figura 5.3, corresponde al proceso de las rutinas de entrenamiento y a la posibilidad posterior de iniciar un ejercicio.

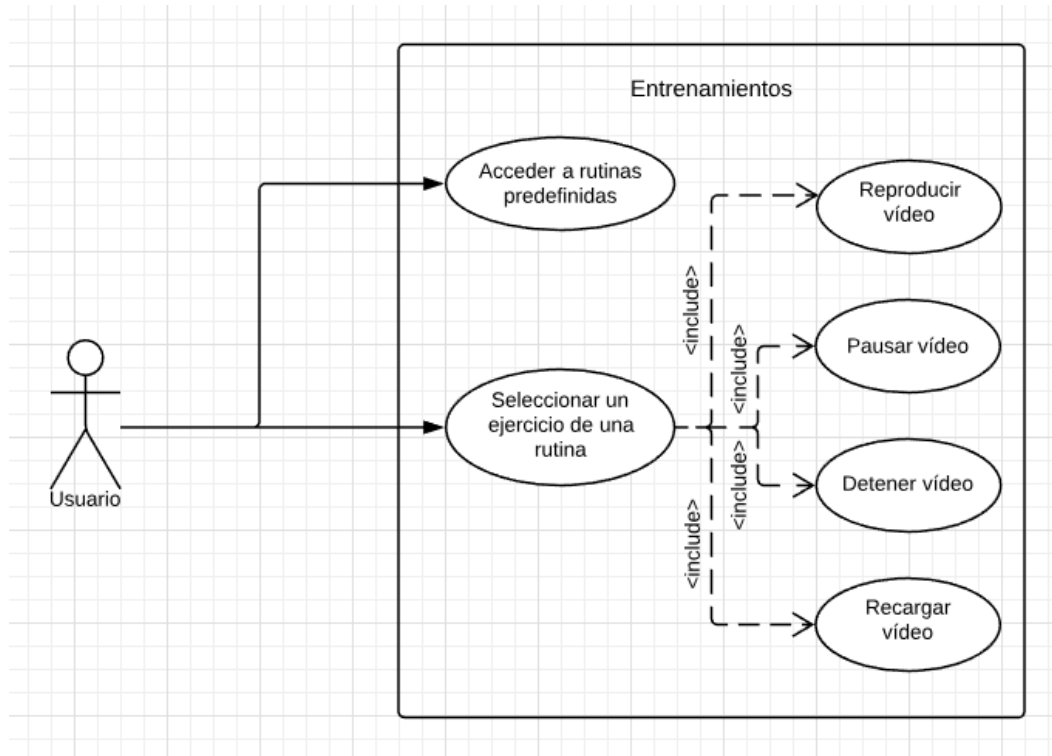


Figura 5.3 Caso de uso I - Entrenamientos.

5.1.2.2 Compartir

En el diagrama mostrado en la figura 5.4 se refleja el uso de la pantalla de compartir de la aplicación, en la cual se pueden enviar archivos a otros usuarios.

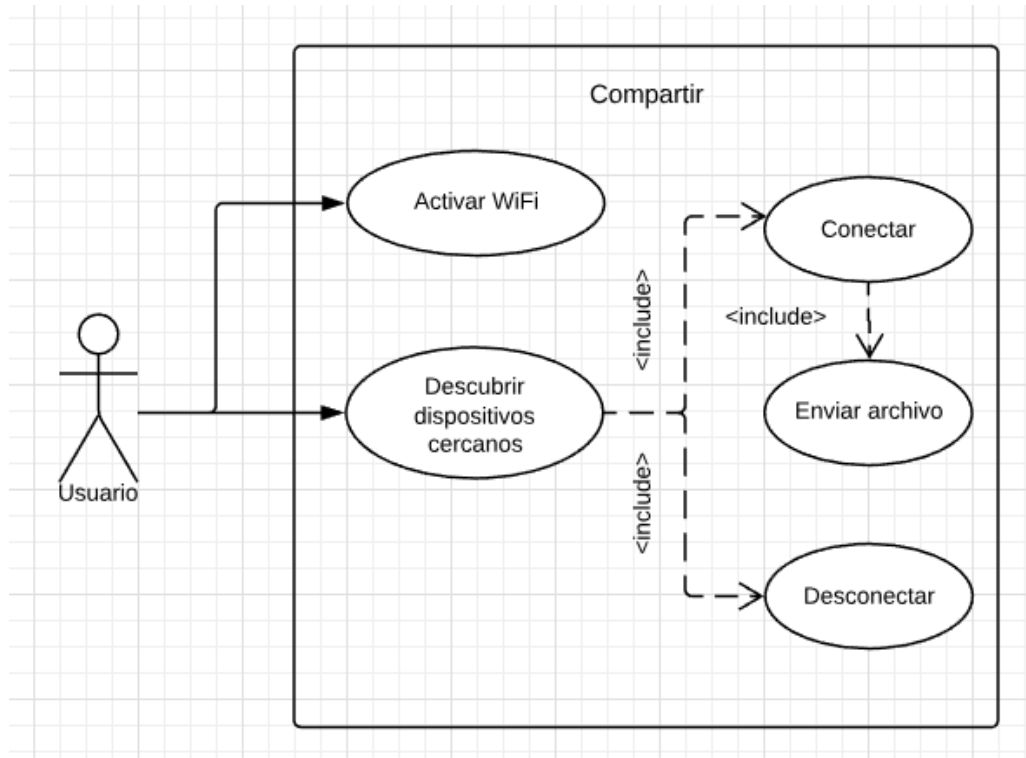


Figura 5.4 Caso de uso II - Compartir.

5.1.2.3 Datos de usuario

El caso de uso mostrado en la figura 5.5 revela el funcionamiento de la parte de datos de usuario, donde dicho usuario puede almacenar su información.

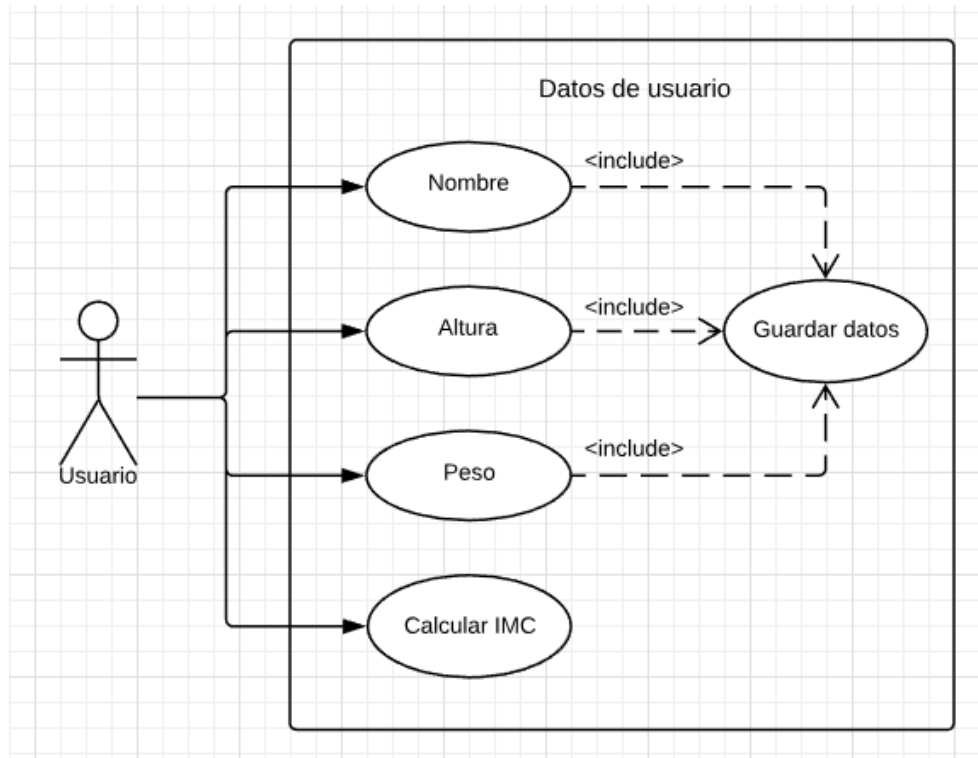


Figura 5.5 Caso de uso III – Datos de usuario.

5.2 Diagramas de clases

El diagrama de clases es un tipo de diagrama de estructura estática que describe la organización de un sistema mostrando sus clases, atributos, métodos y las relaciones entre los objetos. La figura 5.6 muestra el diagrama de clases de esta aplicación. Para una mayor claridad, sólo se han añadido las clases principales y se ha dividido en dos partes.

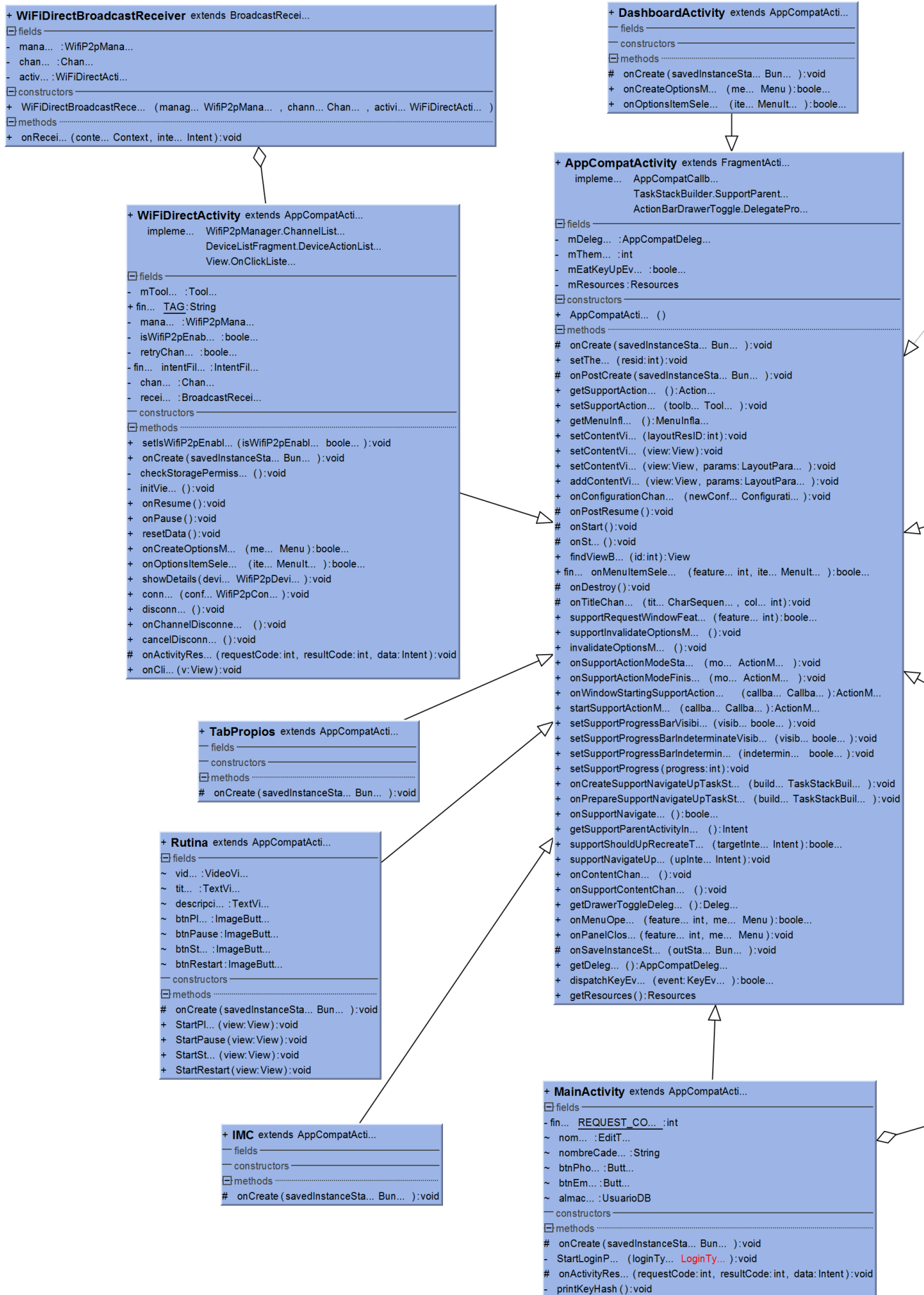


Figura 5.6 Diagrama de clases. Parte 1.

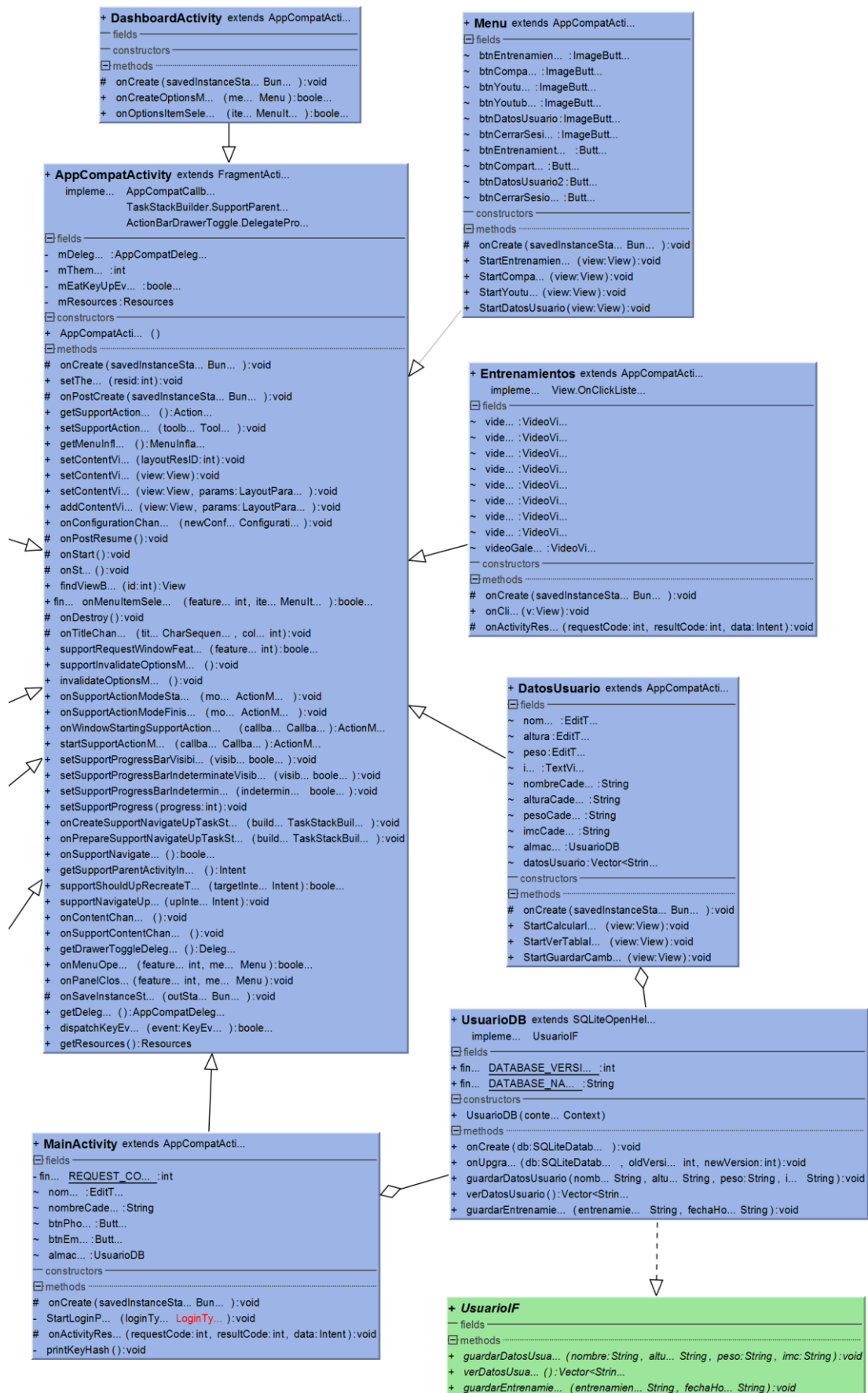


Figura 5.7 Diagrama de clases. Parte 2.

6 CONCLUSIONES

Este sexto capítulo expone las conclusiones a las que se ha llegado tras la realización del presente trabajo, así como los posibles puntos a mejorar del mismo.

6.1 Posibles mejoras

A continuación se exponen algunas de las posibles mejoras del proyecto, las cuales se pueden implementar en futuras líneas de desarrollo:

- Estado de forma física: El usuario podría indicar su nivel de entrenamiento, es decir, indicar si es un deportista principiante, medio o avanzado, con el objetivo de acotar las rutinas de entrenamiento en función de ello. Asimismo, se podría orientar de forma más exacta del número de repeticiones del ejercicio, la intensidad, el tiempo, etc.
- En relación al punto anterior, sería interesante también incluir alguna clase de prueba inicial para que el usuario pudiera evaluar su forma física a través de la aplicación, ejecutando una serie de entrenamientos y analizando parámetros tales como: cansancio, frecuencia cardíaca, sudoración, tiempo consumido, entre otras.
- Historial: Implementar un historial de ejercicios para que el usuario tenga constancia de las rutinas realizadas en entrenamientos anteriores.
- Estadísticas: Una implementación atractiva sería la generación de gráficas y estadísticas, los cuales podrían incluir, por ejemplo, el rendimiento del usuario, el nivel de intensidad, las pulsaciones durante el ejercicio y las calorías quemadas.
- Redes sociales: Dotar a la aplicación de la funcionalidad de compartir por redes sociales tus entrenamientos y estadísticas, para poder competir con amigos y familiares, convirtiéndose así en un estimulante extra para otros usuarios.
- Retos: Incluir desafíos en la aplicación, como un reto de treinta días de entrenamientos, promoviendo así una vida sana y motivando al usuario.

6.2 Conclusiones

Tras todo lo expuesto en el presente documento y una vez finalizadas todas las fases del desarrollo de la aplicación, se pueden plantear las conclusiones obtenidas de la realización de este trabajo fin de grado.

Se planteó un escenario específico con una serie de requisitos concretos a resolver por la aplicación, tales como:

- Desarrollar una aplicación para el sistema operativo Android perteneciente al deporte, con rutinas de entrenamiento y distintos ejercicios físicos para mantenerse en forma.
- Proveer a dicha aplicación con una índole colaborativa, en el cual los distintos usuarios pudieran compartir sus propios entrenamientos con otros deportistas.
- Emplear una interfaz gráfica sencilla a la par que intuitiva, la cual pudiera ser utilizada por todo tipo de público.

La aplicación Android desarrollada en este trabajo satisface la totalidad de los requisitos, creando así una herramienta eficiente y práctica en el ámbito del entrenamiento físico.

Además, me gustaría manifestar mi valoración personal en las siguientes líneas, destacando todo lo que me ha supuesto la realización de este trabajo.

En primer lugar, quisiera resaltar lo gratificante y enriquecedor que supone llevar a cabo un trabajo fin de grado de esta naturaleza, así como la cantidad de esfuerzo necesario para poder realizarlo, desde los esbozos iniciales hasta la finalización completa del proyecto, pasando por los diversos contratiempos que van surgiendo.

He puesto en práctica una gran cantidad de conocimientos adquiridos a lo largo de toda mi formación académica, como el lenguaje de programación Java, la gestión de bases de datos y la programación en Android, entre otros.

También he adquirido multitud de conceptos desconocidos hasta ahora por mí, tanto teóricos como prácticos, como la tecnología usada para compartir archivos a través de WiFi Direct y técnicas de conexiones inalámbricas entre dispositivos.

Para concluir, me gustaría dar de nuevo las gracias a todas las personas que han estado a mi lado durante este largo camino, sin las cuales esto no habría sido posible.

7 ANEXO A: MANUAL DE INSTALACIÓN

Este capítulo hace de guía para la instalación de Android Studio, lo cual es necesario si se quiere modificar el código de la aplicación. Además, se explicará la importación del proyecto y cómo instalar la aplicación en un dispositivo.

Si sólo se desea tener la aplicación en el dispositivo, únicamente será necesario descargarse y ejecutar el apk de la misma. Para ello es necesario tener 17.06 MB de almacenamiento disponible para el correcto funcionamiento de la aplicación. Una vez instalada, se puede utilizar directamente sin necesidad de instalar ningún programa adicional.

Por el contrario, si se desea modificar la aplicación, se requiere la instalación de Android Studio y la posterior importación del proyecto de la aplicación. Para ello lo primero que se debe hacer es descargar la versión más reciente de Android Studio desde la página web principal de desarrolladores de Android, como se muestra en la figura 7.1.

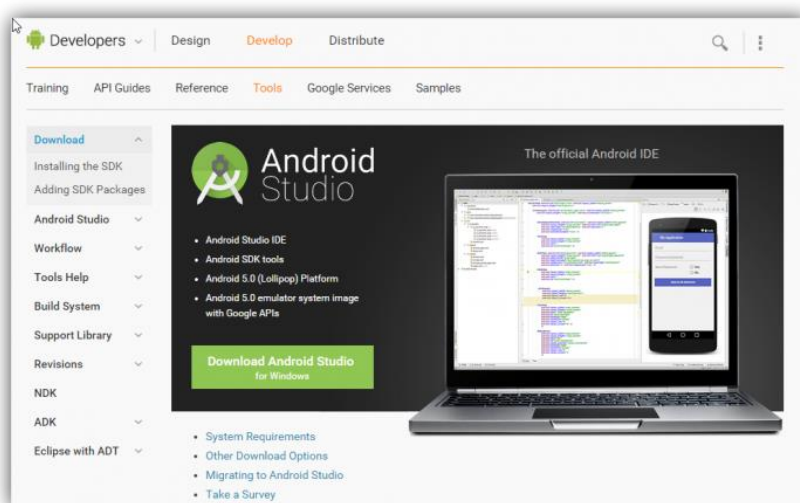


Figura 7.1 Descarga

Una vez descargado el instalador se ejecuta en el equipo para comenzar con la instalación. Lo primero que se verá será el asistente de instalación de Android Studio, mostrado en la figura 7.2.



Figura 7.2 Asistente de instalación.

Siguiendo con el asistente, preguntará por los componentes que se quieren instalar, tal y como se muestra en la figura 7.3.

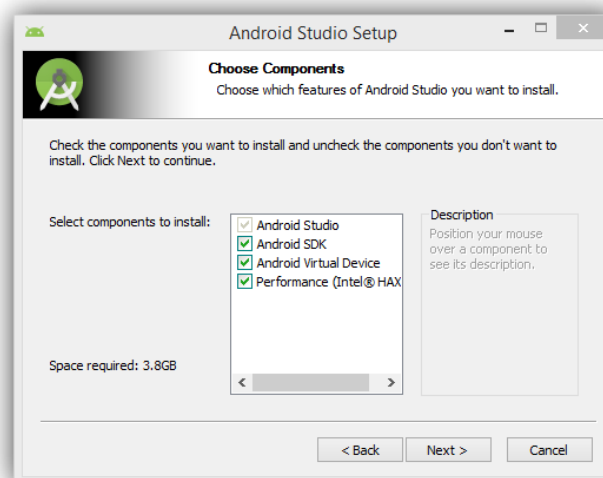


Figura 7.3 Componentes a instalar.

Más adelante se muestra la licencia y a los términos de uso, que se debe aceptar para poder seguir con la instalación. Esto se muestra en la figura 7.4.

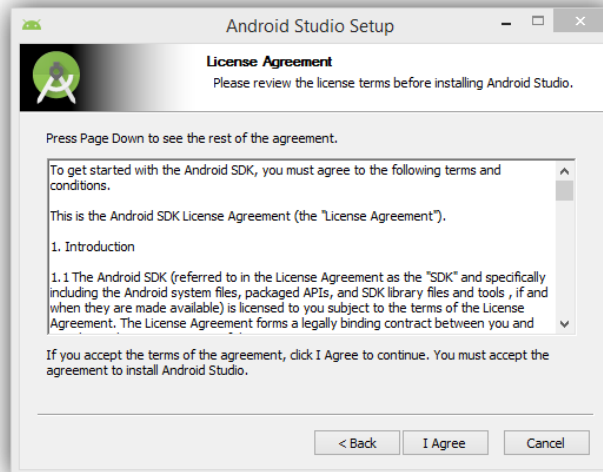


Figura 7.4 Términos de uso.

En el siguiente paso, mostrado en la figura 7.5, se debe elegir la ruta donde se instalará Android Studio. Se deberá elegir una ruta para el programa en sí y otra diferente para instalar el SDK.

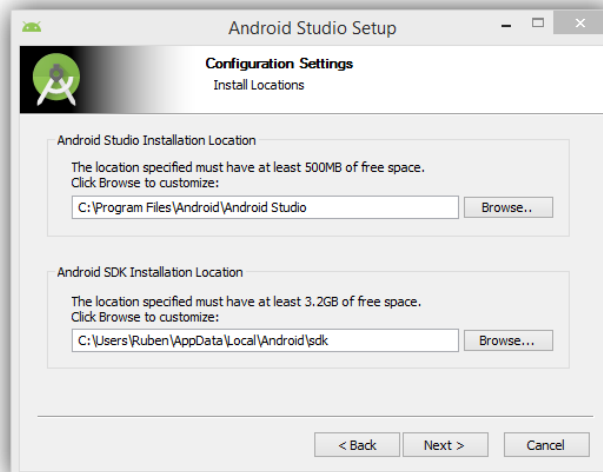


Figura 7.5 Ruta de instalación.

En el siguiente paso el asistente preguntará por la cantidad de memoria RAM que se desea asignar para el uso de máquinas virtuales y emuladores de Android. Esto se muestra en la figura 7.6.

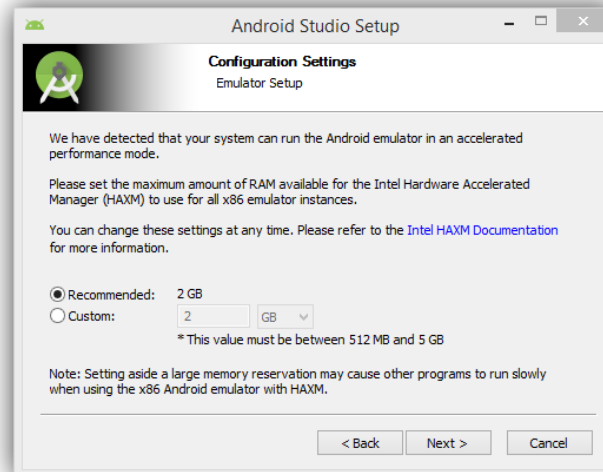


Figura 7.6 Memoria RAM.

Con esto comenzará ya la copia de los archivos al disco duro, como se muestra en la figura 7.7.

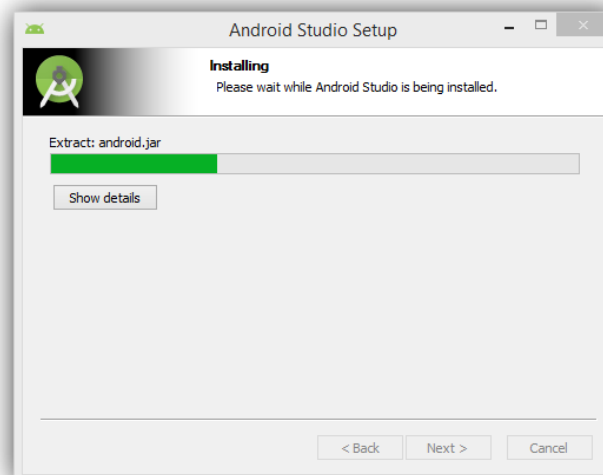


Figura 7.7 Instalando.

Una vez finalice la instalación, Android Studio se conectará a Internet y descargará los elementos del SDK necesarios para funcionar correctamente, tal y como se observa en la figura 7.8.

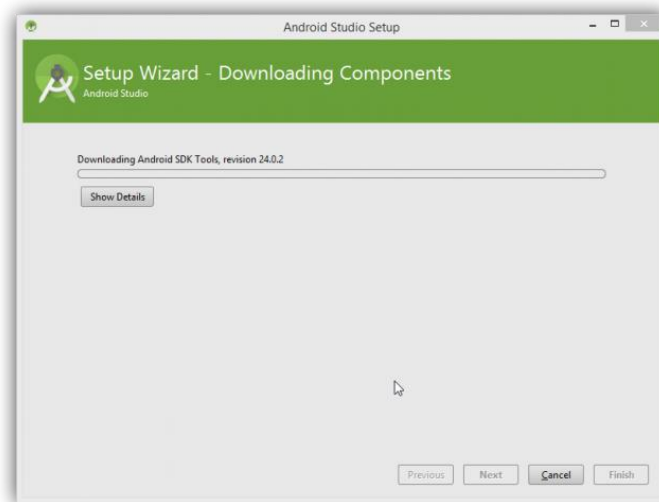


Figura 7.8 Descarga SDK.

Una vez finalice esta copia de datos ya se podrá utilizar el entorno de programación de Android Studio.

Una vez instalado Android Studio, será suficiente con importar el proyecto, tal y como muestra la figura 7.9.

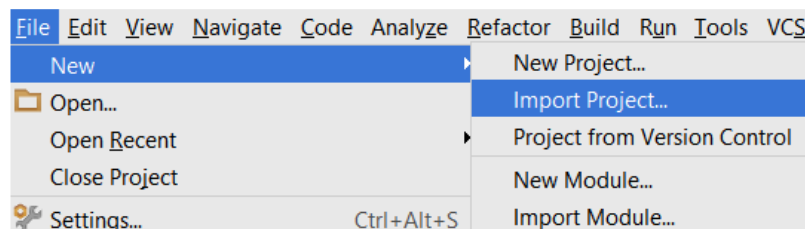


Figura 7.9 Importación del proyecto.

Dado que el proyecto tiene implementadas todas las librerías necesarias para el correcto funcionamiento de la aplicación, no es necesario descargarlas de nuevo ni incluirlas en ninguna carpeta del proyecto.

8 ANEXO B: ACCOUNT KIT

Este capítulo hace de guía para la instalación y configuración de Account Kit, con lo que se consigue tener login a través de SMS o correo electrónico. Se explicarán los pasos para obtener el keyhash, el identificador de la aplicación, el identificador del cliente, la importación del SDK y los meta-data, todo esto necesario para poder configurar Account Kit.

Para empezar, se ingresa en Facebook developers y se crea una aplicación nueva, seleccionando para ello a qué sistema operativo va a ir dirigida la aplicación, tal como aparece en la figura 8.1.

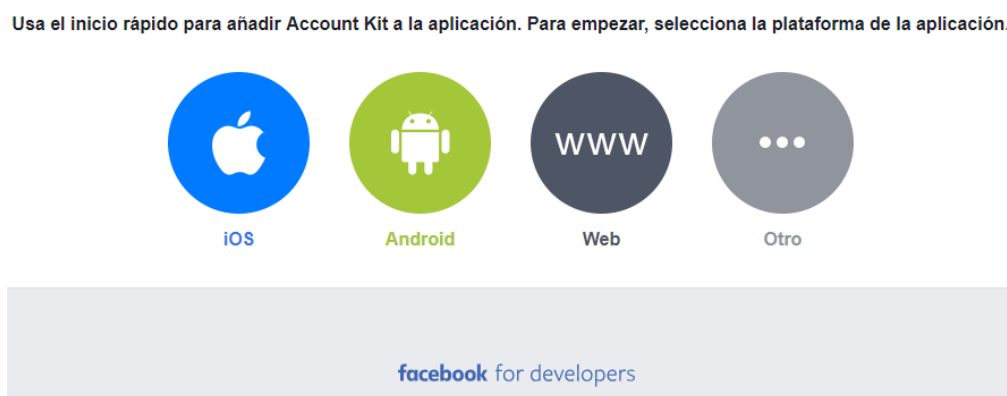


Figura 8.1 Elección del sistema operativo.

Después, es necesario obtener el keyhash. Para conseguirlo, se crea en Android Studio la función mostrada en la figura 8.2 y se inicia la aplicación.


```

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        printKeyHash();
    }

    private void printKeyHash() {
        try{
            PackageInfo info = getPackageManager().getPackageInfo(
                S "edmt.dev.androidfbaccountkit",
                PackageManager.GET_SIGNATURES);
            for(Signature signature : info.signatures)
            {
                MessageDigest md = MessageDigest.getInstance("SHA");
                md.update(signature.toByteArray());
                Log.d( tag: "KEYHASH", Base64.encodeToString(md.digest(), Base64.DEFAULT));
            }

        } catch (PackageManager.NameNotFoundException e) {
            e.printStackTrace();
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
    }
}

```

Figura 8.2 Código para conseguir el keyhash.

Se añade el keyhash y el nombre del paquete de la aplicación en Facebook developers, tal y como se muestra en la figura 8.3.

Android

Inicio rápido ×

Nombre del paquete de Google Play

com.tfg.ejercicios

Nombre de la clase

com.tfg.ejercicios.MainActivity

Hashes de clave

a8siPwofKKPt+k96BRYhEGfOIoY= ×

Figura 8.3 Nombre del paquete y keyhash.

A continuación, se añade el identificador de la aplicación al fichero string de Android Studio, mostrado en la figura 8.4 y en la 8.5.

Figura 8.4 Identificador de la aplicación.

```
strings.xml
Edit translations for all locales in the translations editor.
resources string u
29
30 <string name="FACEBOOK_APP_ID">1936611096669293</string>
```

Figura 8.5 Añadir el identificador en string.xml.

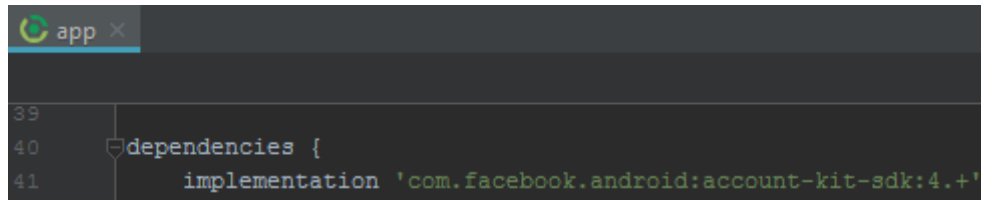
Se hace lo mismo con el identificador de acceso del cliente de Account Kit, lo cual se obtiene en la pestaña Account Kit > Configuración. Además, es necesario tener una clave para el protocolo de login, que no es más que añadir el prefijo “ak” delante del identificador del cliente. Todo esto se muestra en la figura 8.6 y en la 8.7.

Figura 8.6 Identificador de acceso del cliente.

```
strings.xml
Edit translations for all locales in the translations editor.
resources string u
32
33 <string name="ACCOUNT_KIT_CLIENT_TOKEN">4ea802d2f937594cc594a9d936751ee1</string>
34 <string name="ak_login_protocol_scheme">ak4ea802d2f937594cc594a9d936751ee1</string>
```

Figura 8.7 Añadir el identificador del cliente en string.xml.

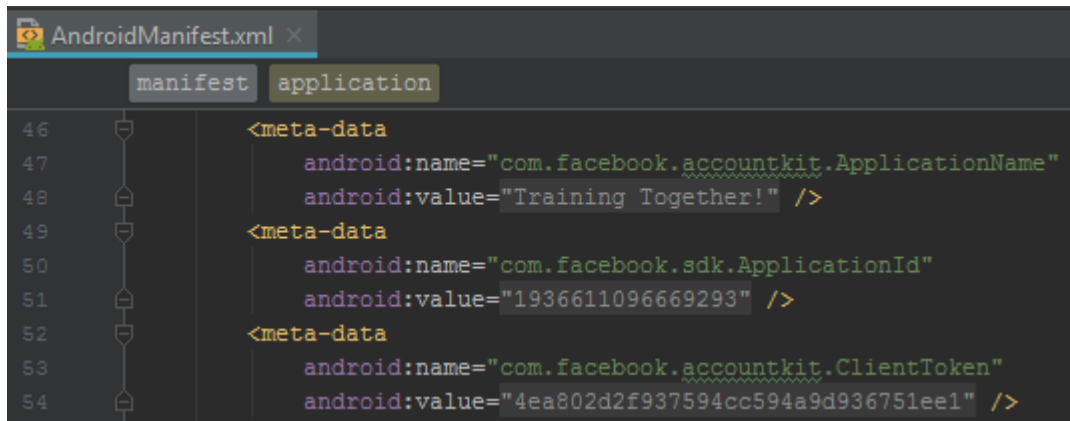
Después, se importa El SDK de Account Kit, como se muestra en la figura 8.8.

A screenshot of an IDE window titled 'app'. The code shows a 'dependencies' block with an 'implementation' entry for 'com.facebook.android:account-kit-sdk:4.+'.

```
39
40 dependencies {
41     implementation 'com.facebook.android:account-kit-sdk:4.+'
```

Figura 8.8 Importación del SDK.

Por último, se añade el meta-data en el AndroidManifest, como se muestra en la figura 8.9.

A screenshot of an IDE window titled 'AndroidManifest.xml'. The 'application' tab is selected. The code shows three <meta-data> entries within the application tag, each with an android:name and android:value attribute.

```
46 <meta-data
47     android:name="com.facebook.accountkit.ApplicationName"
48     android:value="Training Together!" />
49 <meta-data
50     android:name="com.facebook.sdk.ApplicationId"
51     android:value="1936611096669293" />
52 <meta-data
53     android:name="com.facebook.accountkit.ClientToken"
54     android:value="4ea802d2f937594cc594a9d936751ee1" />
```

Figura 8.9 Añadir meta-data en AndroidManifest.xml.

A partir de aquí, ya se tienen todas las claves necesarias para poder continuar con el login de SMS y correo electrónico.

BIBLIOGRAFÍA

[1] Trabajo Fin de Grado: Aplicación Android de rutinas de entrenamientos adaptadas al usuario usando SQLite y JSON, por Miriam Franco Maireles.

[2] Identificadores de acceso - Account Kit - Documentación - Facebook para desarrolladores. (2018). Retrieved from https://developers.facebook.com/docs/accountkit/accesstokens/?locale=es_ES

[3] Revelo, J. (2018). Tutorial De Bases De Datos SQLite En Aplicaciones Android. Retrieved from <http://www.hermosaprogramacion.com/2014/10/android-sqlite-bases-de-datos/>

[4] Reproducción de vídeo desde RAW en Android | Imagina Formacion. (2018). Retrieved from <https://www.imaginaformacion.com/reproduccion-de-video-desde-raw-en-android/>

[5] Android Notes 1: Transparent Status Bar on Fragment and Activity [UPDATED]. (2018). Retrieved from <https://freakycoder.com/android-note-1-translate-status-bar-c2d0dce45aff>

[6] Configure your build | Android Developers. (2018). Retrieved from <https://developer.android.com/studio/build/gradle-plugin-3-0-0-migration>

[7] API Client Library for Java | Google Developers. (2018). Retrieved from <https://developers.google.com/api-client-library/java/>

[8] Console?, H. (2018). How to get the Browser Key for GCM as per the latest Developer Console?. Retrieved from <https://stackoverflow.com/questions/39633044/how-to-get-the-browser-key-for-gcm-as-per-the-latest-developer-console>

[9] Autenticación Sin Contraseña con Account Kit. (2018). Retrieved from <https://code.tutsplus.com/es/tutorials/quick-tip-passwordless-authentication-with-account-kit--cms-28033>

[10] SQLite Tutorial With Example In Android Studio [Step by Step] | Android Database Tutorial In Android Studio: Store Your Data. (2018). Retrieved from <https://abhiandroid.com/database/sqlite>

[11] Wi-Fi peer-to-peer overview | Android Developers. (2018). Retrieved from <https://developer.android.com/guide/topics/connectivity/wifip2p>

[12] SQLite: ejemplo de acceso a base de datos – Academia Android. (2018). Retrieved from <https://academiaandroid.com/sqlite-ejemplo-acceso-a-base-de-datos/>

[13] Socket.IO — Native Socket.IO and Android. (2018). Retrieved from <https://socket.io/blog/native-socket-io-and-android/#>

[14] Getting started with WebRTC for Android — Loopback P2P calls. (2018). Retrieved from <https://vivekc.xyz/peer-to-peer-video-calling-webrtc-for-android-4132fd0ac54>

[15] Android | WebRTC. (2018). Retrieved from <https://webrtc.org/native-code/android/>

[16] Local networking in android – WiFi direct. (2018). Retrieved from <https://androiddevsimplified.wordpress.com/2016/09/13/local-networking-in-android-wifi-direct/>

[17] WiFi direct service discovery in android. (2018). Retrieved from <https://androiddevsimplified.wordpress.com/2016/09/14/wifi-direct-service-discovery-in-android/>

[18] Create P2P connections with Wi-Fi | Android Developers. (2018). Retrieved from <https://developer.android.com/training/connect-devices-wirelessly/wifi-direct>

[19] [closed], H. (2018). How can I transfer files between Android devices using Wi-Fi Direct?. Retrieved from <https://stackoverflow.com/questions/25638335/how-can-i-transfer-files-between-android-devices-using-wi-fi-direct>

[20] How to Transfer Files Using Wi-Fi Pair Connection in Android |. (2018). Retrieved from <https://acadgild.com/blog/how-to-transfer-files-using-wi-fi-pair-connection-in-android>

[21] mayfourth/WiFi-Direct-File-Transfer. (2018). Retrieved from <https://github.com/mayfourth/WiFi-Direct-File-Transfer>

[22] Sending Images over Bluetooth in Android. (2018). Retrieved from <https://kamrana.wordpress.com/2012/05/12/sending-images-over-bluetooth-in-android/>

[23] DB Browser for SQLite. (2018). Retrieved from <https://sqlitebrowser.org/>

[24] Java, S., Node.js, S., PHP, S., Python, S., & It, T. (2018). SQLite Tutorial - An Easy Way to Master SQLite Fast. Retrieved from <http://www.sqlitetutorial.net/>

[25] Revelo, J. (2018). Tutorial De Bases De Datos SQLite En Aplicaciones Android. Retrieved from <http://www.hermosaprogramacion.com/2014/10/android-sqlite-bases-de-datos/>

[26] Usar SQLite en Android Studio. (2018). Retrieved from <https://androidstudiofaqs.com/tutoriales/usar-sqlite-en-android-studio>

[27] Save data using SQLite | Android Developers. (2018). Retrieved from <https://developer.android.com/training/data-storage/sqlite>

[28] Wi-Fi Direct. (2018). Retrieved from https://es.wikipedia.org/wiki/Wi-Fi_Direct

[29] Interactivo, C. (2018). ¿Qué es WiFi Direct y cómo usarlo?. Retrieved from https://parentesis.com/tutoriales/Que_es_WiFi_Direct_y_como_usarlo

[30] ¿Qué es WiFi Direct y para qué sirve? | AndroidPIT. (2018). Retrieved from <https://www.androidpit.es/que-es-wifi-direct>

[31] Santos, E. (2018). Facebook anuncia Account Kit, una utilidad para iniciar sesión en apps sin contraseña. Retrieved from <https://www.genbeta.com/redes-sociales-y-comunidades/facebook-anuncia-account-kit-una-utilidad-para-iniciar-sesion-en-apps-sin-contrasena>

[32] Lacort, J. (2018). Facebook Account Kit: inicios de sesión sin contraseñas. Retrieved from <https://hipertextual.com/2016/04/facebook-account-kit>

[33] Facebook Account Kit: ingresa a apps usando solo tu número telefónico. (2018). Retrieved from <https://www.cnet.com/es/noticias/facebook-account-kit-ingresar-apps-usando-solo-numero-telefonico/>

[34] Facebook para desarrolladores. (2018). Account Kit - Documentación - Facebook para desarrolladores. [online] Available at: https://developers.facebook.com/docs/accountkit?locale=es_ES